UNIVERSIDADE FEDERAL DE SANTA CATARINA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

"UM ESTUDO DA ESPECIFICAÇÃO DE MENSAGEM DA MANUFATURA (MMS) DO PROTOCOLO DE COMUNICAÇÃO PARA AMBIENTES INDUSTRIAIS MAP"

DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL DE SANTA CATARINA

PARA OBTENÇÃO DE GRAU DE MESTRE EM ENGENHARIA ELÉTRICA

ADÃO NAZARENO PINHO

FLORIANÓPOLIS, DEZEMBRO DE 1988

"UM ESTUDO DA ESPECIFICAÇÃO DE MENSAGEM DA MANUFATURA (MMS) DO PROTOCOLO DE COMUNICAÇÃO PARA AMBIENTES INDUSTRIAIS MAP"

ADÃO NAZARENO PINHO

ESTA DISSERTAÇÃO FOI JULGADA PARA A OBTENÇÃO DO TÍTULO DE MESTRE EM ENGENHARIA - ESPECIALIDADE ENGENHARIA ELÉTRICA - E APROVADA EM SUA FORMA FINAL PELO CURSO DE PÓS-GRADUAÇÃO.

Prof. Jean-Marie Farines, Dr.Ing.
Orientador

Prof. José Carlos Moreira Bermudez, Ph.D.
Coordenador do curso de Pós Graduação em Engenharia Elétrica

Banca Examinadora

Prof. Elizabeth Sueli Specialski, M.Sc.

Prof. Jean-Marie Farines, Dr.Ing.

Prof. Joni da Silva Fraga, Dr.

Prof. Manuel de Jesus Mendes, Dr.

ÍNDICE

RESUMO

O desenvolvimento da informática, nestes últimos anos, produziu uma evolução na área de Controle de Processos e Automação Industrial, em direção à utilização de Sistemas Informáticos Distribuídos.

A complexidade destes sistemas exige a adoção de conceitos e metodologias para a concepção confiável das atividades de comunicação. Dentro deste contexto são apresentados o Modelo de Referência para a Interconexão de Sistemas Abertos (RM-OSI) da ISO ("International Standard Organization") e as Técnicas de Descrição Formal (FDT).

Para facilitar a integração dos mais variados equipamentos e dispositivos utilizados na Automação Industrial, a comunidade internacional vem empreendendo grandes esforços de padronização, resultando na definição de uma arquitetura de comunicação para ambientes industriais MAP ("Manufacturing Automation Protocol").

Neste trabalho são apresentadas as características gerais das várias arquiteturas propostas no projeto MAP, com destaque especial para a Especificação de Mensagem de Manufatura (MMS), adotada como protocolo da Camada de Aplicação.

São discutidos os vários conceitos envolvidos neste protocolo, destacando-se, através de uma abordagem formal por Rede de Petri, alguns aspectos do seu comportamento. É também apresentado um exemplo ilustrativo de utilização dos serviços MMS numa Célula Flexível de Usinagem (CFU).

ABSTRACT

In the last years, Informatics in the field of Process Control and Industrial Automation has evoluted towards the utilisation of Distributed Informatic Systems.

The complexity of these systems requires the utilization of concepts and methodologies for reliable conception of communication activities. In this context, the Reference Model for Open Systems Interconection (RM-OSI) of the International Standard Organization (ISO) is introduced, as well as Formal Description Technics (FDT).

With the purpose of facilitating the integration of several equipments and devices used in the Factory Automation, the international society has endeavored to reach high level of standardization, which resulted in a communications architecture standard for manufacturing automation, otherwise called Manufacturing Automation Protocol (MAP).

In this work, the main characteristics of the several proposed architectures in MAP initiative are described, with particular focus on Manufacturing Message Specifications (MMS), which is accepted as the basis of the Application Layer Protocol.

The several concepts of this protocol are discussed, with emphasis to some cases of their behavior, by means of their formal description by Petri Networks. One illustrative example of utilization of the MMS services is also presented in one flexible cell.

CAPÍTULO 1

INTRODUÇÃO

Com o desenvolvimento ocorrido na área de microeletrônica, os equipamentos de computação passaram a ter um maior desempenho, ao mesmo tempo em que reduziram os seus custos, viabilizando a utilização em toda uma gama de aplicações comerciais e industriais, que antes eram anti-econômicas.

Na área de Controle de Processos e Automação Industrial, os Sistemas tradicionais de supervisão e de Controle utilizavam um único computador central, radialmente conectado com vários dispositivos de entrada (aquisição de dados) e saída (atuação no processo). Tais configurações caracterizam-se pela centralização de todas as funções do sistema tanto a nível de controle (filtragem de sinal, algorítmos de regulação, programação de seqüências lógicas, etc...) quanto da interface homem-máquina. Em conseqüência, necessita-se de sistemas computacionais com elevada capacidade de processamento devido ao volume muito grande de informação, resultando num equipamento centralizado de grande porte, de custo elevado e pouco flexível em termos de expansão. Para aumentar a disponibilidade e confiabilidade desse

tipo de sistema, é indispensável a utilização de redundância computacional, com a duplicação do mesmo, tornando-o um sistema de custo inicial muito alto.

A evolução alcançada na tecnologia de integração em larga escala (VLSI) viabilizou, nestes últimos anos, a distribuição das funções dos Sistemas de Controle em vários equipamentos computacionais, interconectados por uma rede de comunicação /Willians 84/. Esta solução apresenta uma melhoria em termos de modularidade do sistema /Enslow 81/ da qual resultam as seguintes características: aumento na flexibilidade facilitando expansões futuras e na disponibilidade, pois a perda de alguns sistemas degrada o funcionamento, mas não o paraliza completamente, podendo haver redundância nos sistemas fundamentais; aumento no desempenho computacional através do aproveitamento da capacidade de processamento distribuído entre as várias unidades do sistema; redução do custo de implantação devido ao processamento junto ao dispositivo controlado, reduzindo as linhas de transmissão; além disso o custo inicial de implantação é pequeno por permitir uma evolução modular, associados a expansão e confiabilidade do sistema.

Os Sistemas Distribuídos vem evoluindo hoje a partir de contribuições obtidas em Arquitetura de Sistemas, Sistemas Operacionais e Linguagens, Protocolos de Comunicação, Engenharia de Software e Algorítmica Distribuída. Em conseqüência, seu campo de aplicação vem crescendo substancialmente substituindo-se em muitos casos a concepção centralizada que prevalecia. Na área de Automação Industrial, o uso de Sistemas Distribuídos torna possível a integração progressiva em direção a uma

Manufatura Integrada por Computador (C.I.M.="Computer Integrated Manufacturing"), das diversas atividades envolvidas durante o ciclo de produção industrial, desde serviços de escritório, administração, projeto de engenharia até o controle da produção, transporte, processamento de materiais e teste de qualidade /Mendes 86/.

Estas diversas atividades são geralmente classificadas segundo uma estrutura funcional hierárquica /Gomide 86, Shapiro 87/; a mais aceita atualmente apresenta os cinco níveis seguintes:

-Gerenciamento:

Envolve atividades situadas no nível da corporação; é o nível mais alto do sistema, onde as informações do mundo externo da companhia e das diversas plantas são coletadas e interpretadas, para a tomada de decisões estratégicas, como gerenciamento, administração e planejamento a longo e médio prazo (anos e meses).

-Planejamento:

O seu escopo situa-se ao nível da planta (fábrica, usina, etc), envolvendo a coordenação de toda a planta e de todos os níveis inferiores. Realiza a gestão da produção a curto prazo (meses e anos) e a otimização do desempenho.

-Coordenação:

Situa-se ao nível de operação das áreas e a tarefa mais importante é a geração de planos de produção para a operação local e a alocação eficiente dos recursos locais. O plano de produção tem uma escala de tempo de semanas e dias, mas deve responder em "tempo real", aos problemas de produção.

-Supervisão:

Situa-se no nível das unidades de operação ("células"). Realiza as tarefas associadas a coordenação e sincronização de um grupo de ferramentas de produção, cooperando para o mesmo trabalho; a escala de tempo é de horas e minutos.

-Controle Direto da Planta:

Está situada no nível da operação do processo. É responsável pelo controle operacional direto da planta, através de máquinas ou equipamentos automáticos, como por exemplo, máquinas ferramentas, linhas de montagem, conversores, motores, robôs, veículos, etc, que precisam ser controlados diretamente por controladores baseados em microprocessadores (8, 16 ou 32 bits). O controle digital direto é realizado por controladores lógicos programáveis, comandos numéricos computadorizados, controladores de robôs, controladores digitais universais e dedicados, etc.. O tempo envolvido situa-se na faixa de minutos e segundos.

Os benefícios trazidos pela integração de todas estas atividades poderão ser viabilizados somente através da garantia de uma comunicação sem erro e sem perda de informação, entre os diversos equipamentos do sistema /LeLann 83b/. O sistema de comunicação deverá ter os mesmos requisitos exigidos para o restante do sistema, como robustez, exatidão, prontidão e flexibilidade /LeLann 83a, Prince 81/.

O requisito da **robustez** provêm de uma composição dos requisitos de confiabilidade e disponibilidade, e visa garantir a reconfiguração sem interrupção de funcionamento, o isolamento rápido de erros e a manutenção rápida.

O requisito da **exatidão** permite a avaliação da integridade do conteúdo de uma informação recebida por uma tarefa através do sistema de comunicação, dado que em sistemas de tempo real é preferível não entregar uma mensagem do que entregá-la com algum erro.

Prontidão é um requisito que garante tempos de atraso da comunicação inferiores a valores pré estabelecidos.

Os requisitos de exatidão e prontidão de um sistema podem ser entendidos como decorrentes das necessidades de tempo real, estando também associados com o requisito de robustez. Em conseqüência, podemos dizer que para aplicações em tempo real é importante que o comportamento de um sistema de comunicação seja determinista, cumprindo desta forma os requisitos de tempo.

Enfim, o requisito de **flexibilidade** indica a capacidade de adaptação e expansão do sistema de comunicação decorrentes de novas necessidades operacionais, substituição de equipamentos (defeitos, novas tecnologias, etc.). A flexibilidade pode ter caracteristicas funcionais, implementacionais, geográficas e computacionais, de acordo com /Lelann 82/.

Estes requisitos estão na base da concepção de Sistema de Comunicação. Entretanto, os benefícios trazidos por esta concepção em ambientes como o industrial, necessitam de esforços no sentido da obtenção e do uso de padrões que assegurem a interoperacionalidade entre equipamentos de diferentes fabricantes e diversas tecnologias.

Neste sentido, podemos citar como de grande importância para os sistemas integrados de produção, os trabalhos feitos pela General Motors, na proposta MAP ("Manufacturing Automation Protocol"), juntamente com a proposta TOP ("Technical Office Protocol"), lançada pela Boeing.

A presente dissertação visa apresentar um estudo de parte da proposta MAP referente a camada de Aplicação, apresentando o protocolo MMS ("Manufacturing Message Specification"), adotado para esta camada e cuja finalidade é interconectar os diversos equipamentos utilizados nos níveis inferiores da estrutura hierárquica do processo de produção industrial.

No capítulo 2 são apresentados alguns conceitos e metodologias para a concepção confiável de protocolos de comunicação, destacando, em particular, o Modelo de Referência para a Interconexão de Sistemas Abertos da ISO ("International Standard Organization") e as técnicas de descrição formal (FDT), utilizados para a especificação das atividades de comunicação.

As várias arquiteturas propostas no projeto MAP são apresentadas no capítulo 3, onde também são vistos os conceitos envolvidos com a Especificação de Mensagens da Manufatura (MMS); um exemplo de como utilizar MMS numa Célula Flexível de Usinagem posicionará este protocolo dentro de um sistema de produção.

O capítulo 4 contém a descrição parcial do protocolo MMS, juntamente com uma abordagem sistemática para sua validação, a partir da análise do modelo obtido com o uso de redes de Petri.

CAPÍTULO 2

CONCEPÇÃO DE SISTEMAS DISTRIBUÍDOS

2.1. Introdução

A complexidade dos Sistemas Distribuídos, particularmente no que diz respeito aos aspectos ligados a comunicação, tornou necessário a definição de metodologias e técnicas que garantem uma operação confiável destes sistemas.

A definição de um modelo de arquitetura estruturada em camadas, chamado Modelo de Referência para a Interconexão de Sistemas Abertos (RM-OSI) da ISO, resultou dos esforços da comunidade técnica internacional e serve atualmente de base para o entendimento da maior parte dos problemas de comunicação. Neste capítulo, serão apresentados alguns dos principais aspectos e conceitos (camadas, protocolos, serviços, interfaces de comunicação) do modelo RM-OSI.

A verificação de aspectos, tais como a consistência lógica da especificação antes da implementação e a conformidade desta com as especificações, tornou necessário o uso de metodologias baseadas em Técnicas de Descrição Formal (FDT) para a obtenção de modelos dos sistemas distribuídos; ferramentas associadas a estas técnicas ajudam o usuário a conceber e implementar tais sistemas. Na segunda parte deste capítulo, serão apresentados aspectos ligados a especificação e verificação de Sistemas Distribuídos a partir da utilização de Técnicas de Descrição Formal.

Neste capítulo será descrito o Modelo de Referência OSI, a divisão da especificação em Especificação de Serviço, Especificação de Protocolos e Especificação de Implementação, além das técnicas formais para estas especificações. Em particular, é feito uma breve descrição das Redes de Petri e das linguagens Estelle e Lotos.

2.2. Aspectos Arquiteturais

2.2.1. Introdução

A necessidade de permitir a comunicação entre equipamentos de tipos diferentes e de fabricantes diferentes sem que seja feita qualquer alteração no "hardware" ou no "software", levou a adoção de um modelo arquitetural. O Modelo de Referência para a interconexão de sistemas abertos OSI ("Open System Interconection") da ISO ("International Standarts Organization") é hoje reconhecido e aceito como base para solucionar problemas de comunicação, compatibilizando os equipamentos entre si e, desta forma, facilitando a concepção de aplicações distribuídas /Mendes 86/.

Este modelo é caracterizado principalmente pela descrição das funções de comunicação, que é realizada através de uma forma abstrata, hierarquizada e dividida em sete camadas, como será visto a seguir.

2.2.2. O Modelo de Referência OSI

2.2.2.1. Arquitetura em Camadas

Devido a complexidade dos aspectos de comunicação nos sistemas distribuídos, foi adotado para o modelo OSI-ISO /ISO 7498/ uma Estrutura Modular em Camadas, de forma a permitir a decomposição lógica do sistema de comunicação em pequenos subsistemas, de menor complexidade e que podem ser desenvolvidos separadamente, o que proporciona uma grande vantagem para a concepção e implementação de tais sistemas /Bochmann 83, Lages 86/. O objetivo de cada camada é fornecer alguns serviços bem definidos à camada imediatamente superior, e utilizar os serviços fornecidos pela camada imediatamente inferior para realizar os seus próprios serviços, sem levar em conta os detalhes de implementação e fornecimento destes serviços.

Cada uma das camadas do modelo é constituída por Objetos Lógicos, denominados Entidades, que executam um conjunto bem definido de funções.

O modelo de comunicação adotado é apresentado como a interação entre entidades chamadas Usuários de Serviço, realizada através de uma outra entidade, o Provedor de Serviço, que utiliza um conjunto de regras para gerenciar a transferência e o formato de mensagens, chamado de "Protocolo" /Day 83/.

Usuário e Provedor de Serviço comunicam-se através das Primitivas de Serviço. A comunicação consiste numa interação estabelecida através da referência comum aos diversos parâmetros da primitiva, aos quais tanto o usuário quanto o provedor podem utilizar.

Esta interação é executada no Ponto de Acesso ao Serviço (SAP), limite comum ente Usuário e Provedor /Vissers 86/. Um SAP é um conceito abstrato que permite a interligação de duas entidades de protocolo das camadas adjacentes (N) e (N+1), e representa os limites internos nos sistemas reais; um SAP(N) é caracterizado por um endereço(N) que identifica a entidade(N+1) que o utiliza. Cada SAP(N) é servido por somente uma entidade(N); uma mesma entidade(N) pode servir a vários SAP(N). Cada SAP(N) é utilizado por somente uma entidade(N+1). Uma mesma entidade(N+1) pode utilizar vários SAP(N).

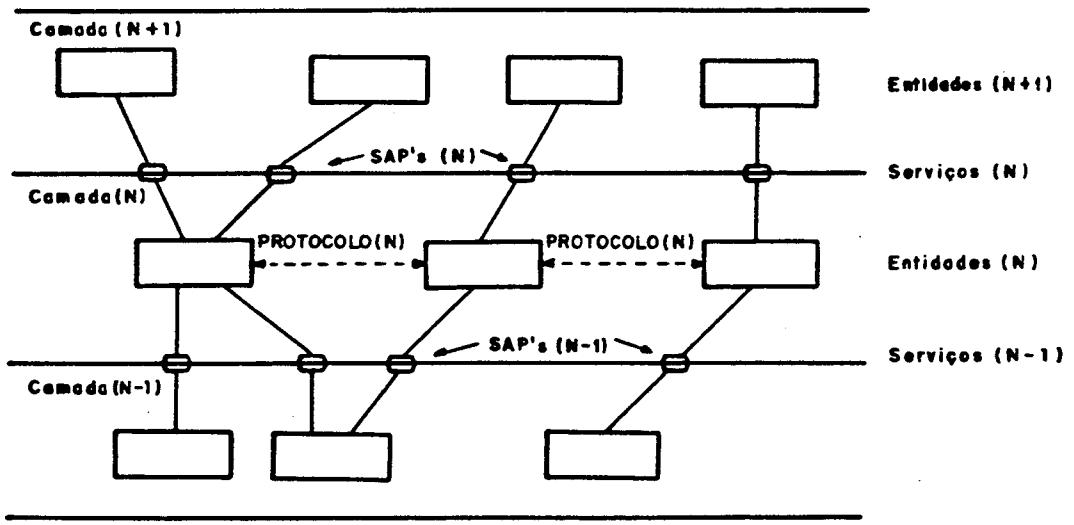


Fig. 1: Modelo de Comunicação

A figura 1 mostra o modelo de comunicação adotado no Modelo de Referência OSI/ISO.

A representação do Provedor de Serviço(N) é baseada na cooperação de Entidades(N), que implementam o protocolo de comunicação transferindo as Unidades de Dados de Protocolo, denominadas PDU(N) ("Protocol Data Unit"), através dos serviços(N-1), que por sua vez possuem entidades que implementam o seu próprio protocolo através dos serviços oferecidos pelas camadas inferiores. O protocolo(N) é visto como uma implementação lógica do serviço(N), à partir do serviço(N-1) /Courtiat 87/. A figura 2 apresenta o modelo do Provedor de Serviço.

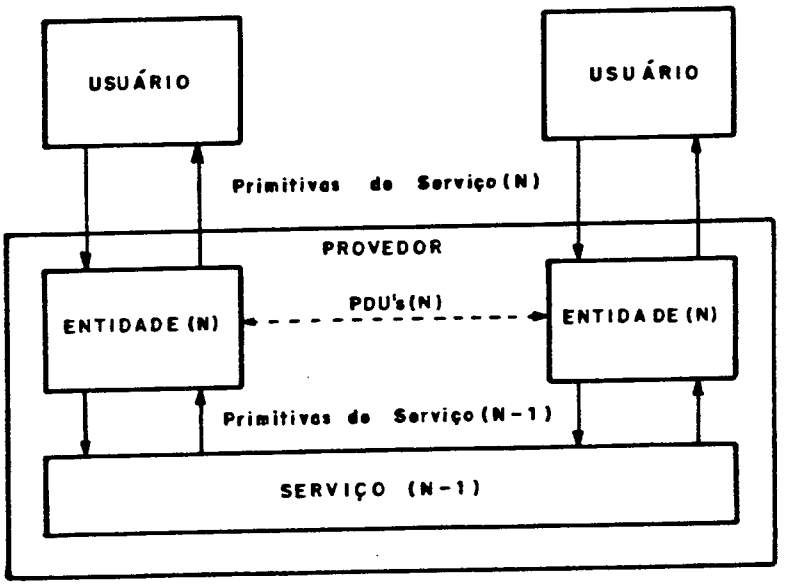


Fig. 2: Provedor de Serviço

Através das Primitivas de Serviço é obtida a representação conceitual da interação entre entidades que permite, além da simples passagem de valores, a negociação e a verificação da informação transferida /Vissers 86/. Estas primitivas não necessitam estar relacionadas diretamente aos elementos do protocolo. Cada implementação pode definir a forma de acesso às primitivas da maneira que achar mais conveniente /Linington 83/: por troca de mensagem ou chamada de procedimento (monitores, semáforos, etc.). As primitivas de serviço admitidas são as seguintes:

- Primitiva de Pedido ("Request") - emitida por um Usuário do serviço para solicitar alguma função;

- Primitiva de Indicação ("Indication") - emitida pelo provedor do serviço para invocar alguma função ou para indicar a um usuário do serviço a ocorrência de um pedido de função por outro usuário do serviço num SAP remoto;

- Primitiva de Resposta ("Response") - emitida pelo usuário do serviço para responder, quando necessário, a alguma função previamente requisitada através da primitiva de Indicação;

- Primitiva de Confirmação ("Confirmation") - emitida pelo provedor do serviço para indicar ao usuário o resultado de alguma função solicitada anteriormente por este.

A figura 3 apresenta as primitivas de serviço.

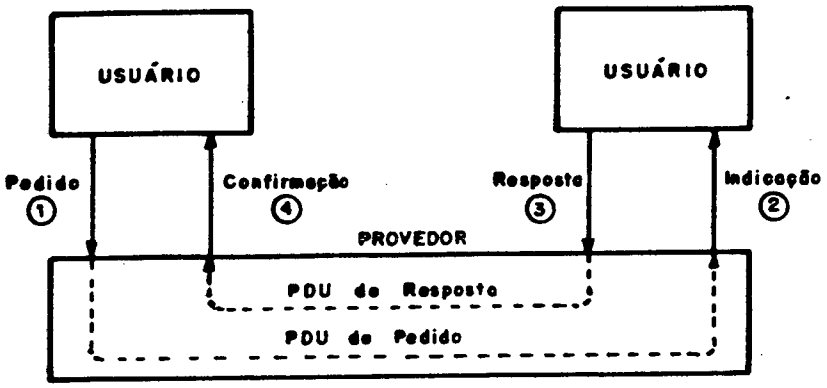


Fig. 3: Primitivas de serviço

O uso conjunto das primitivas de serviço permite classificar os serviços em vários tipos apresentados na figura 4:

- Serviço confirmado, quando a sua prestação envolve os dois pares complementares de primitivas Pedido-Indicação e Resposta-Confirmação;
- Serviço não-confirmado, quando emprega apenas o primeiro par de primitivas;
- Serviço iniciado pelo provedor, quando envolve apenas a primitiva de Indicação.

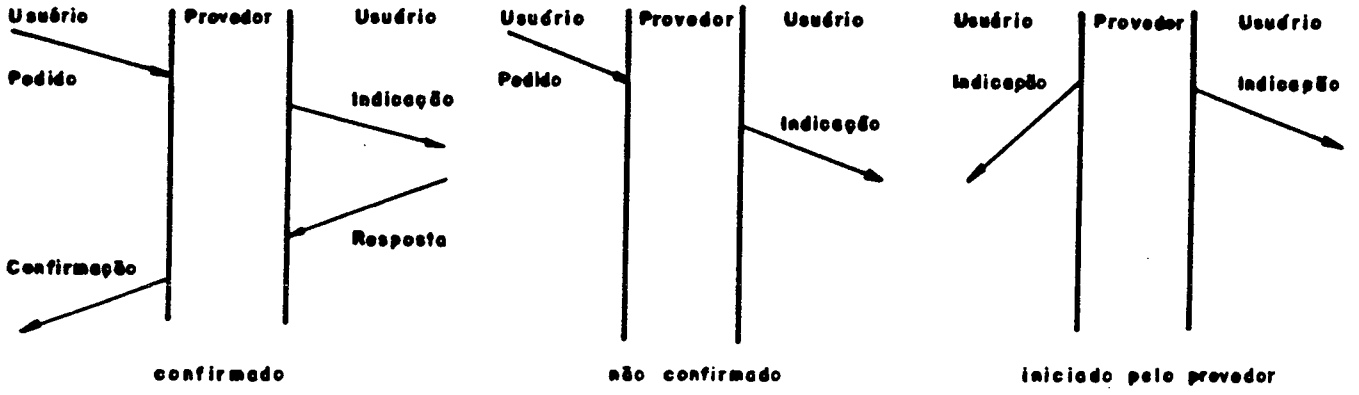


Fig. 4: Tipos de serviços

O conjunto das primitivas de serviço pode também ser subdividido em:

- Primitivas de controle que gerenciam a interação entre entidades (N+1) e (N);
- Primitivas de transferência de dados que permitem a transferência de Unidades de Dado de Serviço, chamada de SDU(N) ("Service Data Unit").

2.2.2.2. Conexão

Um Serviço(N) pode ser orientado a conexão ou sem conexão /Chapin 83/. No primeiro caso, para que duas entidades(N+1) possam trocar informações é necessário o estabelecimento de uma conexão(N). Uma conexão estabelecida é identificada dentro de cada SAP(N) através do seu Ponto Final de Conexão(N), chamada CEP(N) ("(N)Connection End Point") e mostrada na figura 5.

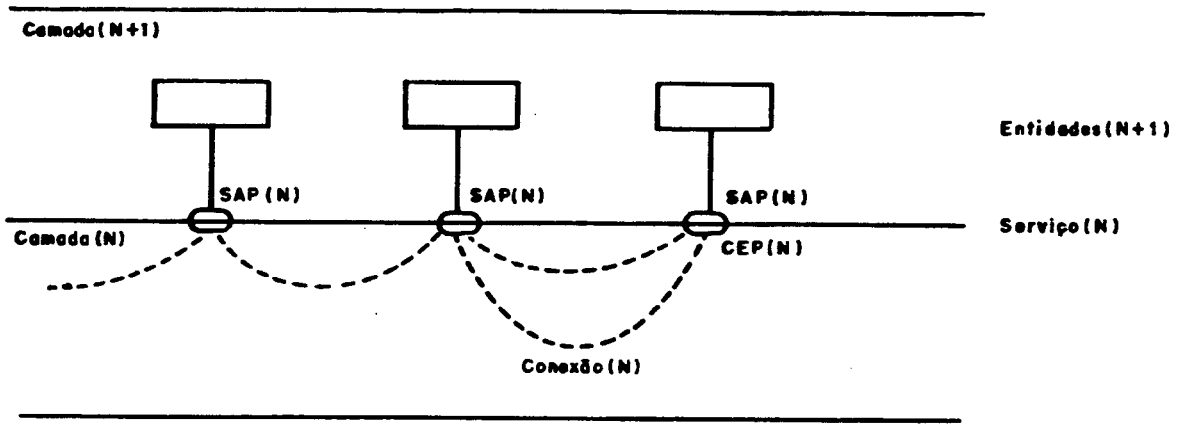


Fig. 5: Conexão

Uma conexão(N) garante para as entidades(N+1) que a informações trocadas serão entregues na mesma ordem que foram emitidas. Um serviço orientado a conexão consiste em três etapas distintas:

- Solicitação e negociação dos recursos e acordos nas regras básicas para sua interação;
- Troca de uma série de unidades de dados de acordo com as regras estabelecidas;
- Encerramento da interação entre as entidades.

Caso o serviço(N) seja um serviço sem conexão, as duas entidades(N+1) podem trocar informações sem necessidade da primeira e da última etapa (abertura e fechamento de conexão). Neste caso não é mais possível garantir nem a entrega nem o seqüenciamento das informações enviadas para a entidade destino.

2.2.2.3. As Camadas do Modelo de Referência OSI

Uma arquitetura com vários níveis de abstração facilita a divisão de responsabilidades para o gerenciamento dos diversos recursos disponíveis, além de proporcionar uma flexibilidade para alterações e mudanças, uma vez que cada nível age como um módulo separado em termos de implementação.

O Modelo de Referência OSI da ISO apresenta 7 camadas cujas características gerais são dadas a seguir:

Camada Física

A Camada Física tem por objetivo estabelecer os procedimentos para a transmissão de mensagens entre os nodos, fixando as características elétricas, mecânicas e funcionais, codificando e transferindo as mensagens através do meio físico /Maclelland 83/.

Camada de Enlace de Dados

A camada de Enlace de Dados tem por objetivo fornecer os serviços necessários para a transmissão de quadros individuais de dados, detectando e, às vezes, corrigindo os erros ocorridos na camada física.

Camada de Rede

A camada de rede tem por objetivo fornecer os serviços necessários para transferir os dados de forma trasparente entre usuários do serviço de rede, controlando o roteamento e controle de congestionamento de mensagens numa rede complexa.

Camada de Transporte

A camada de Transporte tem por objetivo fornecer os serviços necessários para assegurar a integridade dos dados, garantindo que a mensagem seja recebida no destino sem erros e na seqüência correta, sem perda ou duplicação. Ela torna transparente para os usuários destes serviços os detalhes de como as mensagens são transportadas entre a estação emissora e a estação receptora, constituindo-se num protocolo fim-a-fim.

Camada de Sessão

A camada de Sessão tem por objetivo fornecer os meios necessários para que haja a cooperação entre dois processos de aplicação, coordenando a troca de mensagens, de forma a permitir a organização e sincronização do diálogo estabelecido.

Camada de Apresentação

A camada de Apresentação tem por objetivo definir os mecanismos para negociar e relacionar sintaxes de transferência, permitindo a eliminação dos problemas decorrentes dos diversos modos de representação das estruturas de dados utilizadas e da representação do conjunto de ações necessárias ao acesso aos dados, em cada programa do usuário.

Camada de Aplicação

A camada de Aplicação tem como objetivo fornecer os serviços de comunicação diretamente utilizáveis pelos processos de aplicação, constituídos pelos programas dos usuários finais.

A figura 6 mostra a visão global do Modelo de Referência OSI.

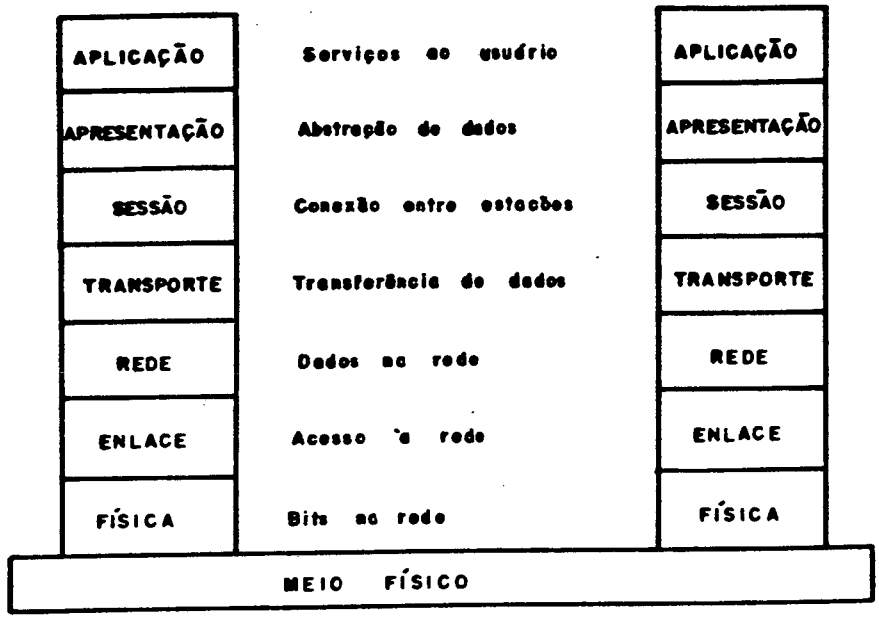


Fig. 6: Modelo de Referência OSI

2.3. Aspectos de Especificação e Verificação

2.3.1. Introdução

As dificuldades encontradas no desenvolvimento de software para Sistemas Distribuídos são idênticas às de qualquer desenvolvimento de software /Courtiat 86, Bochmann 83/, com destaque especial para:

- A necessidade de garantir a compatibilidade entre os diversos componentes heterogêneos do sistema;
- A diversidade de equipes, algumas vezes de organizações diferentes, envolvidas no desenvolvimento de cada um destes componentes;
- A dificuldade de compreensão do comportamento global do sistema, principalmente devido ao paralelismo intrínseco dos seus diferentes componentes.

Em conseqüência, são atualmente admitidas duas formas de descrever as atividades de comunicação nos Sistemas Distribuídos /Vissers 86 e Bochmann 83/:

- O modelo **"observacional"** que corresponde a Especificação de Serviço. Ele define o comportamento do provedor de serviço tal como possa ser observado pelo usuário do serviço; a especificação de um serviço(N) descreve o conjunto de interações possíveis entre a camada(N) e seus usuários(N+1).

- O modelo **"intencional"** que corresponde a Especificação de Protocolo. Ele mostra a estrutura interna do provedor de serviço através do comportamento das entidades de protocolo da camada(N).

A divisão da especificação das atividades de comunicação em Especificação de Serviço e Especificação de Protocolo, diminui o grau de dificuldade inicial de entendimento, apresentando-os sobre estes dois aspectos complementares. Estas especificações poderão servir como base para as atividades de avaliação, verificação, projeto, implementação e teste de conformidade.

Entretanto, para garantir uma definição completa e sem ambigüidade destas atividades de comunicação, deve-se utilizar Técnicas de Descrição Formais (FDT) para especificá-las.

2.3.2. Descrição das Atividades de Comunicação

Como visto anteriormente, as atividades de comunicação podem ser enfocadas sob vários pontos de vista: o do serviço fornecido, do protocolo de comunicação e da implementação deste. A seguir, são apresentados os aspectos principais dos tipos de especificações correspondentes.

Especificação de Serviço:

No modelo arquitetural escolhido, cada camada fornece um conjunto particular de serviços, de forma transparente, aos usuários do nível superior /Zimmerman 80/. A especificação de serviço engloba a descrição das primitivas de serviço, que é a forma pela qual os usuários dos serviços trocam informações com o provedor de serviço, a descrição das regras locais que determinam as possíveis seqüências de execução dessas primitivas, os seus efeitos, e a descrição dos "Pontos de Acesso ao Serviço" (SAP) que é a interface onde essa primitiva pode ser acessada.

Especificação de Protocolo:

A especificação do protocolo define o conjunto de regras que regem as comunicações entre entidades de uma mesma camada /Lages 86/. Os seguintes aspectos são levados em conta:

- **Estabelecimento de um elemento de dado padrão**, através da definição de Unidades de Dados de Protocolo, denominado PDU ("Protocol Data Unit"), que é trocado entre entidades participantes do processo de comunicação; a especificação destes dados vem sendo feita atualmente na Notação de Sintaxe Abstrata ASN.1;

- **Estabelecimento dos caminhos de comunicação**, criando desta forma um meio de comunicação virtual que torna transparente aos usuários as reais características da comunicação;

- **Estabelecimento das convenções necessárias** para a representação da estrutura de dados, do formato de endereçamento das entidades de comunicação, e da seqüência de PDU´s de controle para manter a comunicação, etc..

<u>Especificação de Implementação</u>

A especificação da Implementação é um refinamento da especificação de protocolo, na qual foram acrescidos detalhes necessários a realização de uma implementação, como aspectos da estrutura interna da entidade e os detalhes de comportamento ainda não especificados no protocolo. Além disso, visando implementar as regras do protocolo deve ser incluído um algorítmo, fornecendo os detalhes específicos sobre a interface a ser usada na implementação do SAP e das primitivas de serviço.

2.3.3. As Técnicas de Descrição Formal

2.3.3.1. Introdução

Os protocolos são geralmente descritos através do uso de linguagens naturais. Este tipo de descrição gera uma especificação informal, que apesar de aparentemente facilitar a compreensão, é muitas vezes incompleta, longa, ambígua, inconsistente ou de interpretação variada /Bochman 80/

Para evitar erros de entendimento e para facilitar a compreensão do paralelismo, é necessário utilizar um modelo formal que permitirá a descrição sistemática do comportamento do Sistema Distribuído e a sua posterior validação.

As Técnicas de Descrição Formal (FTDs) tem como papel principal facilitar a especificação e a posterior implementação dos protocolos de comunicação e dos algorítmos distribuídos, tornando-a independente do projetista, não ambígua e compatível /Vissers 83, Souza 87/. Elas servirão a vários propósitos:

- Definir uma especificação de serviços, protocolos, interfaces ou modelo de referência, de forma clara, concisa e não ambígua, que servirá de base para uma cooperação entre os projetistas dos vários componentes do sistema e de ponto de partida para as outras etapas de desenvolvimento;

- Permitir a verificação da consistência lógica do projeto através da análise da especificação quanto a sua exatidão, eficiência, perfeição (integridade), etc.

- Permitir o desenvolvimento de suportes para implementações menos sujeitas a erros (ou mais adequadas);

- Permitir a validação da implementação do ponto de vista da conformidade e da consistência com a especificação.

2.3.3.2. O Modelo de Referência OSI/ISO e as FDT´s

No que diz respeito ao Modelo de Referência OSI da ISO, vimos que são normalmente utilizados dois níveis de abstração: a Especificação de Serviço e a Especificação de Protocolo. A Especificação de Serviço descreve o comportamento de uma camada através da descrição de sua visibilidade externa, do ponto de vista do usuário. Esta descrição é feita através da especificação do conjunto de seqüências válidas de interação que podem existir na interface entre o provedor e o usuário dos serviços oferecidos pela camada. A Especificação de Protocolo descreve como os serviços são oferecidos, através da descrição do comportamento lógico de cada uma das entidades de protocolos que constituem a camada. Portanto, é natural que se utilize Técnicas de Descrição Formal "orientadas a seqüência de eventos" para a especificação de serviço e técnicas "orientadas a estados" para a especificação de protocolos. Porém, geralmente é

necessário, devido a necessidade de se fazer a validação/verificação do protocolo, utilizar-se a mesma técnica de especificação para os serviços e protocolos /Courtiat 87/.

A utilização de uma técnica de descrição formal deve permitir a representação abstrata, dos conceitos abstratos definidos dentro do Modelo de Referência da ISO, como primitivas de serviço, SAPs, CEP e interações inter-camadas, e dos conceitos mais concretos que precisam ser descritos de uma forma mais precisa, como as regras do diálogo de um protocolo, a representação das restrições temporais, a codificação de PDU´S e as regras de segmentação/concatenação de PDU´S, dentre outras.

Existem diversos modelos formais, que permitem representar os mecanismos de comunicação em Sistemas Distribuídos conforme definidos no Modelo de Referência da ISO, sendo que podemos citar como as principais os modelos de máquinas de estado finito, Redes de Petri, abordagem algébrica, gramáticas formais, tipos de dados abstrato e lógica temporal.

Neste trabalho apresenta-se com mais detalhes as Redes de Petri, que serão posteriormente utilizadas na descrição e verificação do protocolo MMS e as duas linguagens Estelle e Lotos. As linguagem Estelle e Lotos estão em processo de padronização pela ISO (Organização Internacional de Normalização) e são destinadas ao uso nas descrições formais dos protocolos normalizados.

Estelle é uma linguagem baseada sobre máquinas de estado finito extendida, acrescida de uma representação de dados tipo Pascal. Lotos é uma linguagem baseada no formalismo algébrico CCS que descreve um protocolo na forma de uma equação do comportamento, acrescido de uma representação de dados na forma de tipos abstratos algébricos.

2.3.4. Redes de Petri

2.3.4.1. Princípios Gerais

A Rede de Petri é um modelo formal adaptado à descrição do paralelismo e da comunicação, que tem uma representação gráfica com nodos de dois tipos: os lugares que representam "condições" e as transições que representam "eventos"; arcos direcionados interligam os nodos. A ocorrência de uma condição é representada através da colocação de uma ficha no lugar que a representa; o conjunto de lugares que possuem fichas define o estado da Rede de Petri.

O disparo de uma transição corresponde a ocorrência de um evento e é mostrado na figura 7.

A Rede de Petri permite construir, de maneira progressiva e modular, através da interconexão dos modelos das suas diferentes partes, o modelo global do protocolo à fim de efetuar uma posterior análise /Ayache 85/.

Entretanto, o modelo da Rede de Petri ordinária é limitado, por não permitir as representações:

- De certas partes do protocolo, tais como dados;

- De um número arbitrário de mensagens pendentes, que devem ser recebidas na mesma ordem em que são enviadas;

- De alguns protocolos com um número ilimitado de topologias permitidas /Merlin 79/.

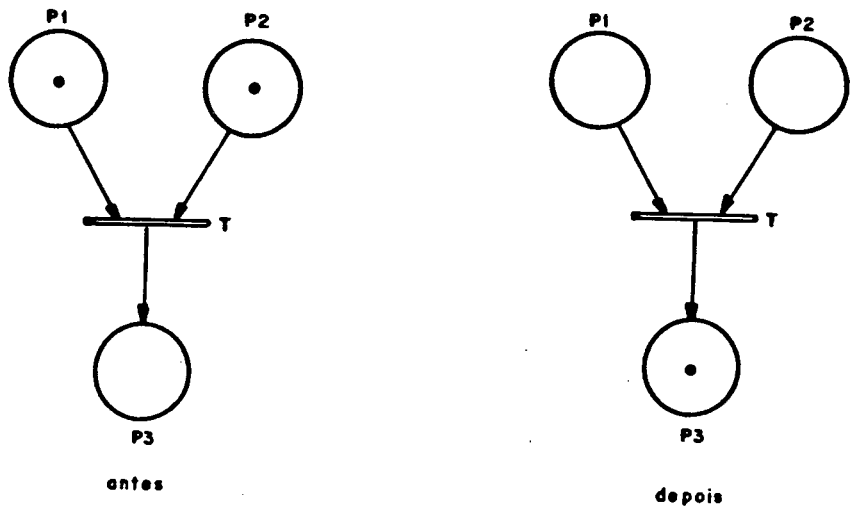


Fig. 7: Disparo de uma Transição

Contudo, é um modelo mais abrangente que a máquina de estado finito, pois permite representar protocolos que tenham um número infinito de estados.

2.3.4.2. Tipos de Redes de Petri

A principal limitação da Rede de Petri é o crescimento excessivo do número de estados, resultando em grafos pouco legíveis para protocolos complexos. Para resolver estes problemas, diversas extensões e generalizações foram propostas a partir do modelo básico que levaram a representações mais compactas. Entretanto a generalidade alcançada dificulta a análise. Dentre destas extensões propostas, a partir da Rede de Petri ordinária, podemos destacar principalmente a Rede Predicado/Ação, as Redes Coloridas, as Redes Predicado/Transição, as Redes de Petri numéricas e as Redes com Temporização.

Rede Predicado/Ação /Keller 76/

Os modelos de Rede de Petri Predicado/Ação, também chamados de Redes de Petri etiquetadas ou interpretadas, permitem que a descrição do sistema seja decomposta em uma parte de controle e uma parte de dados. A parte de controle é descrita por meio de uma rede de Petri, geralmente salva, que exprime os aspectos de sincronização da especificação. A cada transição é associado um predicado e uma ação que atuam sobre as variáveis do sistema, que representam de maneira explícita a parte de dados da especificação. O predicado deve ser satisfeito para que uma transição possa disparar e fazer com que a ação possa ser executada de maneira indivisível. Na modelização de protocolos, os

predicados são geralmente recepções de PDU(N) ou de primitivas(N) e as ações são os envios de PDU(N) ou de primitivas(N), além das ações internas a um módulo. A notação utilizada para cada transição etiquetada é "When P(x) do A(x)", ou de forma simplificada: Predicado/Ação ou ainda utilizando a notação CSP (E? Mi/Ej! Mj).

Redes de Petri Coloridas /Jensen 81/

Na rede de Petri clássica não é possível distinguir as fichas, pois as mesmas não possuem identidade; em conseqüência, tipos distintos de recursos devem ser modelados por diferentes lugares, tornando rapidamente o modelo ilegível. Dentro de uma Rede de Petri colorida, cada ficha possui uma cor que identifica a informação contida; a descrição das fichas coloridas que serão produzidas ou consumidas é dada por uma função associada aos arcos. /Sunshine 82 e Courtiat 87/

Rede Predicado/Transição /Genrich 79/

As redes Predicado/Transição, associam um predicado a cada transição da rede de maneira a especificar a estrutura das operações e as relações existentes entre as fichas. Nas Redes Predicado/Transição cada ficha representa um indivíduo que vai evoluir dentro da rede, simplificando a representação do ponto de vista gráfico.

Rede de Petri Numérica /Billington 87/

A utilização de uma rede de Petri colorida para definir os aspectos de controle, juntamente com extensões que permitam diferenciar, para cada lugar de entrada de uma transição, as condições de sensibilização das regras de disparo, permite chegar a um modelo mais completo que as redes anteriores. Este tipo de rede é conhecida como rede de Petri Numérica e é extremamente bem adaptada para a especificação do comportamento de Sistemas Distribuídos, devido a concisão e a abrangência alcançada.

Redes de Petri com Temporização /Diaz 87/

A descrição correta do comportamento de uma entidade de protocolo, e a análise do desempenho de funcionamento, passa pela introdução de temporizações como parte do modelo de descrição. A rede de Petri com temporização permite representar os aspectos de tempo, associando a cada transição um intervalo de tempo contado a partir da sua habilitação, durante o qual a transição pode disparar.

2.3.4.3. Validação de protocolos utilizando Rede de Petri

A análise das propriedades das Redes de Petri são divididas em duas classes:

- As **propriedades gerais** (limitada, viva, reiniciável) que devem ser satisfeitas para todo protocolo. São analisadas pelas técnicas de análise dinâmica através da evolução da rede (construção de um grafo de marcações acessíveis, por exemplo) e dependem da marcação inicial.

- As **propriedades específicas**, que devem ser obtidas para cada tipo específico de protocolo e dependem do sentido particular dado aos diferentes elementos (lugares e transições da rede) do modelo global. São analisadas usualmente pelas técnicas de análise estruturais e dependem exclusivamente da topologia da rede, não sendo influenciada pela marcação inicial.

a - Propriedades gerais:

São as propriedades que devem ser verificadas e são necessárias para o perfeito funcionamento de qualquer protocolo.

Limitada:

Uma rede de Petri é dita limitada quando, para uma dada marcação inicial e qualquer que seja a seqüência utilizada para disparar as transições, o número de fichas em qualquer lugar permanece limitado. É a primeira propriedade que deve ser satisfeita. Uma rede de Petri não limitada implica num número infinito de estados do protocolo correspondente, o que impossibilita a sua implementação, ou na existência de seqüências que aumentam os recursos disponíveis, o que também não é viável.

Vivacidade:

Uma rede de Petri é viva, quando a partir de cada estado acessível, qualquer outro estado é acessível. É a segunda propriedade desejada para o modelo global do protocolo e implica em diversas propriedades no protocolo:

- **Ausência de bloqueio mortal**, ou seja, não há estado terminal, indicando que a transmissão de mensagem é sempre possível. Pode ocorrer quando não há mensagens em trânsito no meio virtual e todas as entidades envolvidas na comunicação estão em estados que esperam a recepção de alguma mensagem;

- **Ausência de recepção não especificada;** esta situação pode ocorrer quando o meio virtual é modelado como uma fila FIFO (primeiro a entrar - primeiro a sair), não alterando a ordem em que as mensagens foram transmitidas. Assim, pode ocorrer um bloqueio quando a mensagem a ser retirada da fila não corresponde a mensagem a ser recebida, no estado em que a entidade se encontra, fazendo com que fiquem retidas dentro do meio virtual;

- **Ausência de interações não executáveis;** esta situação ocorre quando não há bloqueio na rede, porém, algumas transições não se tornam sensibilizadas, não podendo, portanto, ser disparadas. Podem constituir um erro de concepção de protocolo, quando representam eventos ligados diretamente ao fornecimento do serviço. Porém, podem ser uma consequência do modelo escolhido para interconectar as entidades do protocolo através dos serviços fornecidos pelas camadas inferiores.

Reiniciável:

Uma Rede de Petri é reiniciável, quando a partir de cada marcação acessível é possível retornar a marcação inicial. É a terceira propriedade que um modelo de protocolo deve satisfazer, correspondendo ao fato de que o funcionamento de um protocolo é geralmente cíclico (repetitivo).

b - Propriedades Específicas:

As propriedades especificas obtidas do modelo em Rede de Petri permitem verificar a correção parcial e a evolução do protocolo correspondente.

b.1 Correção Parcial do Protocolo

A verificação da Correção Parcial do protocolo pressupõe o conhecimento de uma especificação do serviço desejado sob a forma ou de um conjunto de asserções a verificar sobre o grafo de marcações (estados) ou de um modelo autômato de estado finito.

No primeiro caso, a correção é provada se as asserções propostas podem ser verificadas diretamente sobre cada estado do grafo de acessibilidade, ou através da obtenção e interpretação dos **invariantes de lugar**, que representam a existência de componentes conservativos no modelo global.

No segundo caso, a prova é realizada através da equivalência entre o modelo global que representa os serviços efetivamente fornecidos pelo protocolo e o modelo de serviço desejado. Para isto é utilizado o método da projeção /Courtiat 84/, que consiste em ter uma visão abstrata do modelo global do protocolo, considerando as primitivas de serviço como únicos eventos visíveis através da interface entre o usuário do serviço e o provedor.

O **Método da Projeção** consiste das seguintes etapas: /COURTIAT 87/

- Dividir todos os eventos associados as transições da rede em dois grupos:

  . Um grupo de eventos visíveis, que correspondem as primitivas do serviço, e que devem aparecer na projeção obtida para o serviço realizado pelo protocolo;

  . Um grupo de eventos internos (invisíveis), os quais não deverão aparecer na visão abstrata; as transições relativas a estes eventos são denominadas transições-lambda;

- Gerar um autômato reduzido a partir da Rede de Petri que representa o modelo global do protocolo, partindo do grafo de marcações acessíveis, eliminando as transições-lambda; este autômato representa o modelo do serviço fornecido realmente pelo protocolo;

- Comparar o autômato reduzido do modelo global, obtido pelo método de projeção com o autômato correspondente à especificação do serviço a ser realizado; a equivalência destes dois autômatos, do ponto de vista das linguagens resultantes, permite verificar a comformidade do modelo global do protocolo implementado com o serviço desejado.

b.2 Evolução do Protocolo

A verificação da evolução do protocolo está associada a procura e interpretação das invariantes de transição do modelo global, que indicam os ciclos repetitivos na rede.

Desta forma, é possível verificar o acontecimento de um determinado evento acessível pelo usuário do serviço, a partir da constatação de sua participação em algum invariante de transição. Também é possível detectar a existência de interbloqueamento, que ocorre quando há um ciclo repetitivo de eventos não visíveis ao usuário.

Através da verificação destes dois tipos de propriedades, é possível provar a correção total do protocolo, pois o primeiro garante a correção parcial durante a evolução do mesmo e o segundo a evolução efetiva.

2.3.5. Linguagem ESTELLE

Estelle é uma linguagem da ISO /ISO DIS 7185/ que é utilizada para a descrição formal de sistemas distribuídos e/ou concorrentes, e em particular de especificações de protocolo e de serviço /Courtiat 86, Budkowski 87/.

A linguagem Estelle é uma técnica de descrição formal (FDT), baseada sobre o modelo de máquina de estado finito extendida, usando um conjunto de extensões da linguagem de programação Pascal, em particular, no que diz respeito ao modelo de representação dos dados.

Uma especificação em Estelle é composta por um conjunto de módulos, que se comunicam entre si, contendo uma descrição bastante completa e precisa, de modo que o comportamento da especificação seja não ambíguo, ou que as ambigüidades (não determinismo) sejam explícitas. As interações de cada módulo com o ambiente são especificadas através da descrição cuidadosa de todas as características do comportamento não-determinista e de suas três grandes partes componentes: a definição de módulo, a estrutura do sistema e a definição do tipo canal.

Módulo é o mais alto nível de abstração dentro do modelo Estelle, e é constituído em duas partes:

"header": que fornece a visibilidade externa do módulo;

"body" : que define o comportamento interno do módulo.

O comportamento de um módulo é dado por um modelo de transição de estados (autômatos), definido por um conjunto de entradas, saídas, estados e transições, que são extendidas para dar a força e a concisão de uma linguagem de programação. Este comportamento interno é visto externamente através de seus pontos de interação.

Ponto de Interação (IP) é uma interface abstrata de um módulo, usado para a troca de dados (interações) com outros módulos. Esses pontos de interação, são enlaces bidirecionais que permitem interações de entrada e saída independentes e assíncronas. As entradas recebidas de outros módulos são colocadas em filas FIFO, consideradas infinitamente grandes, podendo existir uma fila individual para cada IP ou o compartilhamento da fila em comum com outros IP's.

A **Estrutura** do sistema é a parte da especificação que descreve as instanciações dos vários módulos comunicantes, que trocam mensagens (interações) através de pontos de interação, que são interligados através de um canal.

O conceito de **Canal**, de grande importância na estruturação dos módulos, torna a especificação de um módulo independente do ambiente. Isto é conseguido fazendo com que os módulos incorporem somente as próprias ações e os canais incorporem as interações entre os diversos módulos de um sistema com seu ambiente.

2.3.6. Linguagem LOTOS /Brinksma 86, Bolognesi 87, Leão 87/

Lotos, como Estelle, é uma linguagem utilizada para a descrição formal de sistemas distribuídos e/ou concorrentes, que está em desenvolvimento na ISO, e é baseada no conceito de ordenação temporal de primitivas de interação. É essencialmente uma linguagem derivada do CCS, com pequenas extensões.

O conceito de ordenação temporal de primitivas de interação é caracterizado por definir o comportamento de uma parte da especificação somente através da descrição e ordenação de interações com o seu ambiente. Estas interações modelam apenas a visibilidade externa do módulo, proporcionando um nível de abstração da implementação real, desejável nas linguagens de especificação.

LOTOS especifica todos as possíveis seqüências de informações de entrada e saída (primitivas de serviço) que são observáveis nas suas interfaces através de um conjunto de operadores de "ordenação temporal".

Para aumentar o poder de expressão da linguagem, além de manter a integridade e proteção das informações, acrescentou-se a Lotos uma linguagem de definição de Tipos de Dados Abstratos, conhecida como ACT ONE; assim, um processo somente poderá acessar dados através de funções especialmente fornecidas para o tratamento destes dados.

2.4. Conclusão

Neste capítulo, vimos que a adoção do Modelo de Referência OSI da ISO é extremamente útil para resolver os problemas de concepção de programas de comunicação em sistemas distribuídos. A adoção de uma estrutura hierárquica em camadas com vários níveis de abstração, possibilita a separação em funções, permitindo assim a divisão de responsabilidades para o gerenciamento dos diversos recursos disponíveis, além de proporcionar uma flexibilidade para alterações e mudanças, uma vez que cada nível age como um módulo separado em termos de implementação.

Dentro deste modelo, as camadas altas Aplicação, Apresentação, Sessão e Transportes possuem protocolos altamente consistentes com os programas dos usuários finais, estando diretamente relacionadas com os serviços do nível de aplicação; a camada de Rede trata com o roteamento das mensagens entre aplicações comunicantes; as camadas de Enlace de Dados e Física estão relacionadas com os aspectos da comunicação física de mensagens entre nodos adjacentes.

A divisão da especificação de um protocolo de comunicação em Especificação de Serviço e Especificação de Protocolo, permite representar dois aspectos complementares da problemática de comunicação nos sistemas informáticos. Assim um serviço da camada(N) pode sempre ser descrito como resultado da ação combinada do serviço oferecido pela camada(N-1) e do protocolo da camada(N).

Discutiu-se também a importância do uso de Técnicas de descrição formal (FDT´s) para representar as especificações, por serem necessárias para verificar a correção das especificações e assegurar que as implementações sejam corretas e compatíveis, qualquer que seja o projetista, além de serem essenciais para todas as fases do desenvolvimento.

Apresentou-se mais especificamente a Rede de Petri e as técnicas de utilização desta na verificação de protocolos, por ser a técnica de descrição formal a ser utilizada neste trabalho.

CAPÍTULO 3

A ARQUITETURA MAP

3.1. Introdução

O esforço de padronização, no sentido de permitir a interligação de equipamentos heterogêneos em ambientes industriais, está concretizando-se hoje a partir do projeto MAP ("Manufacturing Automation Protocol") /Kaminski 86/, iniciado pela GM para a área da fábrica, e do projeto TOP ("Technical Office Protocol"), iniciado pela Boeing para a área administrativa da empresa /Farowich 86/.

Atualmente, a arquitetura de comunicação MAP está sendo aceita internacionalmente como forma de integrar os mais diversos equipamentos e dispositivos computadorizados utilizados na automação industrial, como controladores programáveis, robôs, máquinas de comando mumérico computadorizada, controladores de campo, etc, independentes do fabricante /Mendes 87/.

Neste capítulo serão apresentadas as várias arquiteturas propostas no projeto MAP, descrevendo mais particularmente a camada de Aplicação da qual será destacado o protocolo MMS ("Manufacturing Message Specification").

3.2. Arquiteturas MAP

3.2.1. Arquitetura MAP Completa

A arquitetura MAP completa é baseada no Modelo de Referência OSI da ISO, onde conceitos específicos voltados para a automação da manufatura são introduzidos em cada uma das sete camadas do modelo.

Na versão 3.0 provisória, os protocolos adotados para as diversas camadas do modelo ISO/OSI são os seguintes: /MAP3.0/

Camada 7 - Elemento do Serviço de Controle da Associação (ACSE), Transferência, Acesso e Gerenciamento de Arquivos (FTAM), Especificação de Mensagem da Manufatura (MMS) e serviços de diretório;

Camada 6 - A Unidade Funcional Núcleo do Protocolo de Apresentação da ISO;

Camada 5 - A Unidade Funcional Núcleo do Protocolo de Sessão da ISO com operação duplex;

Camada 4 - O Protocolo de Transporte classe IV da ISO;

Camada 3 - O Protocolo de Rede sem conexão da ISO;

Camada 2 - O protocolo IEEE 802.2 classe 1/classe 3;

Camada 1 - IEEE 802.4 "Banda larga" ou IEEE 802.4 "Banda portadora" fase coerente.

Quanto ao gerenciamento de rede, as funções básicas são as seguintes: monitoração, controle, configuração, determinação de problemas e recobrimento de erros.

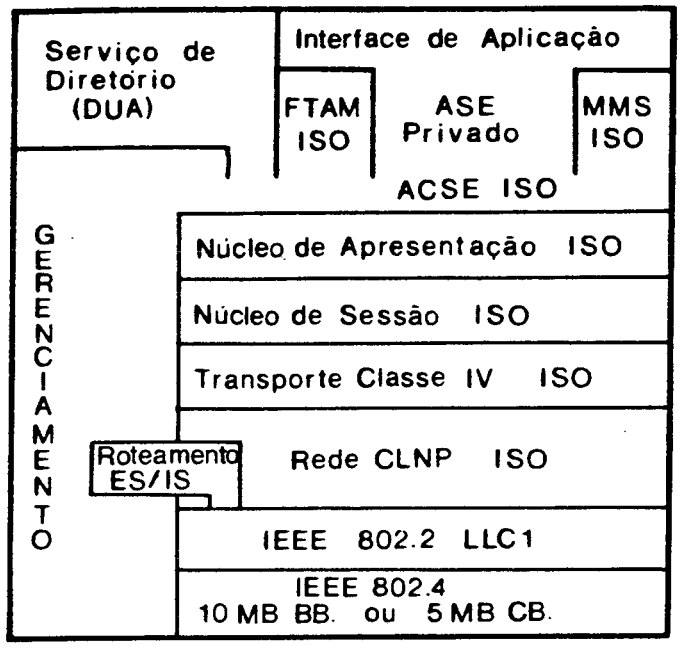


Fig. 8: Arquitetura MAP completa

A seguir são apresentadas de forma mais detalhada as diversas camadas da arquitetura MAP completa /Jayasumana 85, Mendes 86/, que podem ser vistas na figura 8.

Camada 1: Física

Esta camada tem por objetivo fixar as características mecânicas e elétricas para a transmissão de mensagens no meio.

O MAP, para esta camada, baseou-se na norma IEEE 802.4 (passagem de permissão em barramento ou "Token-Bus"). A topologia em barramento tem como vantagem um menor custo de

instalação (fiação) e de expansão; além disso possibilita facilidades para o roteamento e um melhor controle de erros de terminal, pois não possui nodos intermediários.

O meio de transmissão adotado, na arquitetura MAP, é o cabo coaxial, possibilitando uma alta qualidade de transmissão, com taxas de até 300 Mbits e distâncias superiores a vários quilômetros. Devido às características de propagação do sinal, o cabo tem que ter a sua extremidade fechada com um "terminador", com impedância de onda para evitar reflexão do sinal.

A proposta MAP permite a utilização de dois tipos de transmissão física:

- Transmissão "banda larga" ("Broadband"), com taxa de 10MBps, modulação FSK ("Frequency Shift Keying") duobinária e alocação "Mid-Split" nos dois canais de emissão e recepção;

- Transmissão "banda portadora" ("CarrierBand"), com taxa de 5MBps, modulação fase coerente com canal único.

A classe "Banda larga" foi adotada, principalmente devido a existência, nas empresas, de instalações industriais de televisão por cabo, o que diminui o investimento inicial da instalação. Além disso, este tipo de transmissão propicia uma fraca atenuação do sinal, baixa interferência eletromagnética e possibilita uma taxa de transmissão muito alta, através da existência simultânea, no mesmo meio físico, de vários tipos de transmissões, tais como processamento de voz, vídeo, circuito fechado de televisão, teleconferência e outros.

Entretanto, deve-se destacar que, devido a transmissão em "banda larga" ser unidirecional, é necessário instalar um dispositivo modulador/demodulador no final de cada cabo, denominado "Head-end".

Este tipo de transmissão foi adotado pelo MAP para implementar a rede principal ("backbone") que é instalada de acordo com os padrões industriais de televisão por cabo (CATV), de amplo domínio nos meios técnicos.

Para os níveis inferiores de transmissão na fábrica, o MAP adotou a classe "banda portadora", que suporta um único canal bidirecional, sem a necessidade do dispositivo "Head-end", tornando a instalação mais simples e barata. Entretanto, as sub-redes assim constítuidas são restritas a distâncias inferiores a 1000m, aceitando no máximo 32 estações.

Camada 2: Enlace de Dados

A camada de Enlace de Dados tem como papel tornar o meio físico transparente aos usuários desta camada, garantindo a transmissão de dados, sem erros, entre nós da subrede.

O projeto MAP adotou, para o Enlace de Dados, o modelo IEEE 802, que subdivide esta camada em subcamadas de Controle de Enlace Lógico (LLC-"Logical Link Control") e controle de Acesso ao Meio (MAC-"Medium Acess Control").

Subcamada MAC

Esta subcamada tem por objetivo realizar a interface com o meio físico e tornar transparente à subcamada de Controle de Enlace Lógico o método de Controle de Acesso (CSMA/CD,Passagem de permissão em anel ou passagem de permissão em barramento).

Num meio de transmissão em barramento, compartilhado por diversos usuários, é necessário um método de controle de acesso escolhido entre uma das duas classes de métodos: acesso controlado, acesso aleatório.

O projeto MAP adotou para esta subcamada a norma IEEE 802.4 já citada, que tem um método de acesso controlado e descentralizado. Segundo este método, um usuário só poderá transmitir quando possuir a "permissão". Após a transmissão das mensagens por este usuário, a "permissão", que se constitui numa mensagem especial, deverá ser passada para o usuário seguinte, o que equivale a existência de um anel lógico.

O método adotado é considerado adequado para uso em aplicações de tempo real, devido ao suporte e esquemas de prioridades, e sobretudo ao seu comportamento determinista que permite o cálculo preliminar dos tempos de transmissão, exceto nos casos de falha física.

Subcamada LLC

O objetivo desta subcamada é propiciar o estabelecimento de conexão, o controle de fluxo, recuperação de erros, o endereçamento da estação remota, etc.

Dos três tipos de serviços existentes na norma IEEE 802.2: sem conexão e sem reconhecimento (tipo 1), com conexão e com reconhecimento (tipo 2), sem conexão e com reconhecimento imediato (tipo 3), o projeto MAP adotou para esta sub camada o controle de enlace do tipo 1, para evitar uma grande sobrecarga computacional, uma vez que é prevista a implementação das funções de recuperação de erros, contrôle de fluxo e de seqüenciamento, na camada de Transporte.

Devido a necessidade de se alcançar maior segurança de transmissão e implementação de respostas rápidas por parte das estações que não possuem a "permissão", o MAP 3.0, na sua última proposição, adotou também os serviços do tipo 3, que é um protocolo sem conexão, tipo datagrama, mais simples que o tipo 2 e mais completo que o tipo 1 por implementar o reconhecimento imediato; entretanto, continua sem medidas prévias de construção de conexão, sem controle de fluxo e recuperação de erros.

Camada 3: Rede

Esta camada objetiva a transferência de informação fim-a-fim, de tal forma que as mensagens possam circular entre redes distintas sem preocupação com os aspectos específicos de cada uma.

O MAP adotou para esta camada, a estrutura por subcamadas, descritas na norma ISO/DIS 8648, que são o Protocolo de Acesso a Subrede (SNAcP), o Protocolo de Convergência Dependente da Subrede (SNDCP) e o Protocolo de Convergência Independente da Subrede (SNICP), suportando apenas os serviços de Rede no modo sem-conexão (CLNS-"Conection Less-mode Network Services").

Camada 4: Transporte

O objetivo desta camada é tornar transparente aos usuários os detalhes de como os dados são trocados através da rede, garantindo uma comunicação isenta de erros.

O MAP adotou para os serviços e protocolos de Transporte a classe 4 da norma ISO DIS 8072 e 8073, que é a mais sofisticada, permitindo detecção e recuperação de erros, multiplexagem e controle de fluxo dos pacotes enviados através de uma conexão estabelecida entre as entidades comunicantes.

Camada 5: Sessão

O objetivo da camada de Sessão é sincronizar e organizar os dados transferidos no sistema de comunicação.

MAP está definindo, para esta camada, o protocolo ISO 8326/8327, adotando apenas as Unidades Funcionais seguintes:

- Unidade Funcional do Núcleo, que realiza os serviços básicos de início e término de conexão de Sessão, e também dos serviços de transferência de dados normais;
- "Unidade Funcional Duplex", que permite a administração do diálogo entre as entidades de Apresentação do tipo "FULL-DUPLEX", ou seja, nas duas direções simultaneamente;
- "Unidade Funcional de Ressincronização", fornecendo um suporte, ao nível de Apresentação com a inserção de pontos intermediários de sincronização no fluxo de dados, possibilitando retornar a estados anteriores, no caso da ocorrência de erros na comunicação.

Camada 6: Apresentação

Esta camada tem como objetivo resolver os problemas relativos a representação dos dados em cada estação, preocupando-se apenas com o aspecto sintático da transferência de informação.

MAP adota para esta camada a Unidade Funcional "Núcleo da Apresentação" da norma ISO DIS 8823 e ISO DIS 8824, que é a Notação de Sintaxe Abstrata ASN.1, juntamente com a correspondente regra de codificação para sintaxe de transferência ISO DIS 8825; são também previstas a inclusão de outras unidades no decorrer do tempo.

Camada 7: Aplicação

O objetivo desta camada é fornecer os serviços de comunicação diretamente utilizáveis pelos programas dos usuários. Diversos serviços que são muito freqüentemente encontrados nas aplicações dos usuários, estão sendo definidos para esta camada, como por exemplo:

- Elemento de Serviço de Controle da Aplicação (ACSE="Association Control Service Element"), definido na norma da ISO 8649/2 e 8650/2, que se ocupa do controle da associação, através do estabelecimento, término ou aborto das associações estabelecidas entre os Processos de Aplicação;

- Transferência, Acesso e Gerenciamento de Arquivos (FTAM="File Transfer, Access and Management"), segundo a norma ISO DIS 8571/1, 2, 3 e 4, ocupando-se da transferência de arquivos de dados entre Processos de Aplicação, permitindo acesso a arquivos completos e suportando atributos;

- Serviços de Diretório que visam facilitar referências a objetos na rede, de acordo com a norma ISO DP 9594/1.

- Especificação de Mensagem da Manufatura (MMS "Manufacturing Message Specification"), que é o principal Elemento do Serviço de Aplicação (ASE) adotado na camada de Aplicação da arquitetura MAP, o qual se preocupa com os aspectos de comunicação entre dispositivos programáveis na fábrica, definido nas normas ISO DP 9506/1 e 2.

3.2.2. Arquitetura MAP/EPA

Com a estrutura de sete camadas proposta pela arquitetura MAP não é possível obter um tempo de resposta rápido para uso em controle de processos e redes com restrições de tempo. Para resolver estes problemas, a arquitetura MAP completa foi extendida de forma a prover, para alguns nodos, um caminho alternativo para satisfazer os requisitos de tempo de resposta.

Esta arquitetura alternativa é denominada MAP/EPA (Arquitetura de Desempenho Avançado) e pode ser considerada como tendo uma arquitetura dupla, conforme apresentado na figura 9.

Esta arquitetura dupla, é composta de todas as sete camadas que compõe a arquitetura MAP completa (compatível OSI) e uma arquitetura alternativa, baseada no projeto PROWAY, com o fim de diminuir a sobrecarga computacional apresentada no MAP completo,

permitindo desta forma, alcançar tempos de resposta compatíveis com as exigências de tempo de aplicações industriais.

O lado EPA da arquitetura é composto apenas da camada de Aplicação, que necessita implementar apenas os serviços MMS, e da camada de Enlace de Dados e Física, não seguindo, portanto, por inteiro o Modelo de Referência OSI da ISO.

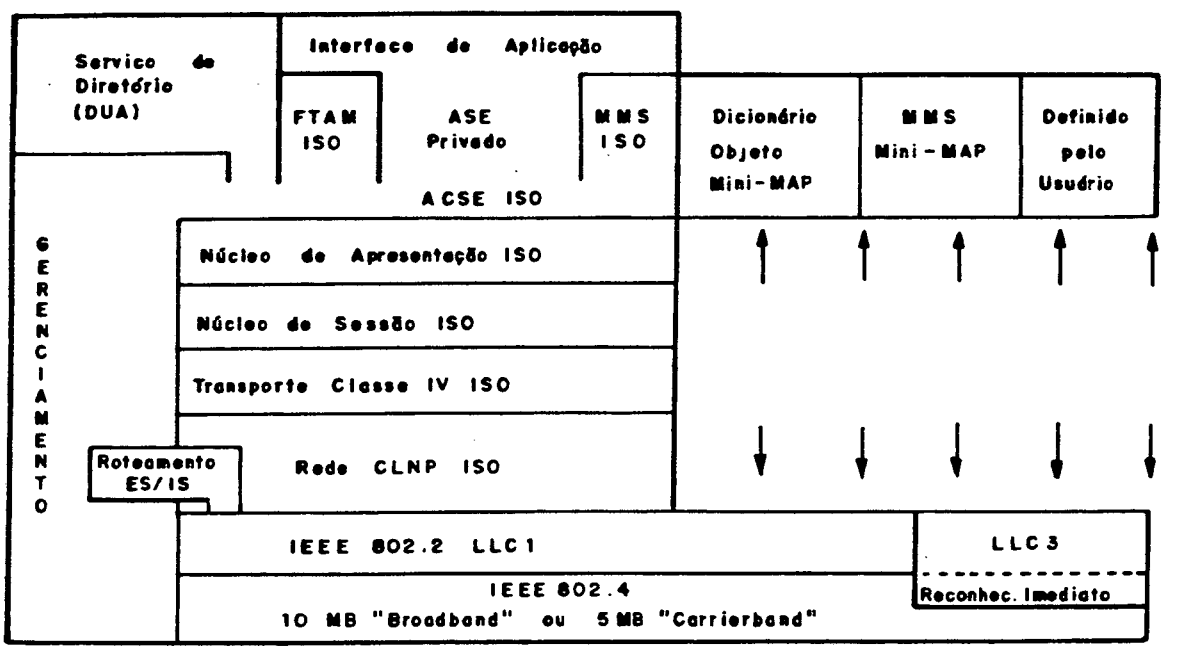


Fig. 9: Arquitetura MAP/EPA

3.2.3. Arquitetura Mini-MAP

Alguns nodos não fazem uso do lado de 7 camadas da arquitetura MAP/EPA, exigindo apenas o lado EPA, suportando, portanto, apenas as três camadas do protocolo: Física, Enlace de Dados e Aplicação. Estes nodos não são compatíveis ISO/OSI e não poderão comunicar-se fora do seu segmento local sem um "Gateway". São denominados nodos mini-MAP e não podem coexistir com segmentos MAP.

A figura 10 apresenta este tipo de arquitetura, onde pode se observar as ausências das camadas intermediárias, fazendo com que a camada de Aplicação acesse diretamente a camada de Enlace de Dados, que pode ser o do tipo LLC 1 ou LLC 3.

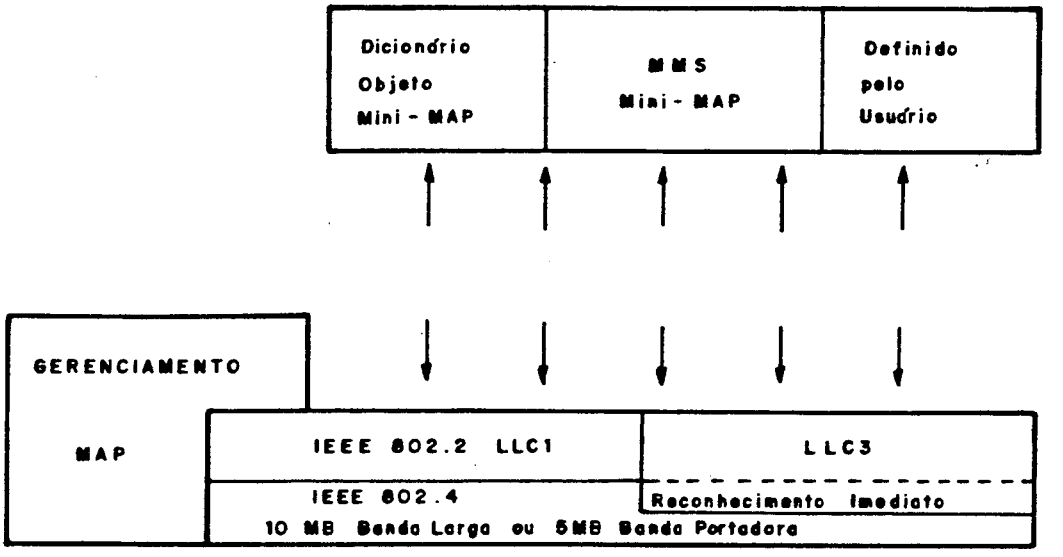


Fig. 10: Arquitetura mini-MAP

3.2.4. Características das Arquiteturas MAP/EPA e Mini-MAP

Os serviços fornecidos pelos nodos mini-MAP são os mesmos que os do lado EPA de um nodo MAP/EPA. Há também a possibilidade, na arquitetura mini-MAP de se utilizar "nodos passivos". Estes nodos são estações que não participam do protocolo de passagem da ficha, entretanto, eles podem responder a pedidos de outros nodos através de mecanismos de resposta imediata. Podem ser utilizados no nível controle local em dispositivos do tipo sensores.

3.2.4.1. Características Gerais

As arquiteturas EPA e mini-MAP são projetadas para existir em um ambiente específico, onde é preciso obter um tempo de resposta rápido para uso em aplicações de controle distribuído. Estas arquiteturas possuem as seguintes características:

- Respostas rápidas para mensagens curtas de alta prioridade;
- Alta disponibilidade do meio (taxa muito baixa de erros e um mínimo de retransmissão);
- Facilidade de manutenção e de expansão;
- Baixa prioridade a qualquer tráfico não necessário às funções de controle principal;

- Facilidade de conexão à rede principal da fábrica;

- Controle de acesso no sentido de manter a segurança do sistema;

- Métodos de redundância para componentes da rede, à fim de manter alta disponibilidade;

- Requer pequeno título para identificação, ou mesmo nenhum suporte, a partir do servidor de Diretório.

3.2.4.2. Comparação com a Arquitetura MAP completa

O mini-MAP e o MAP/EPA operarão usando qualquer tipo de camada Física permitidas para o MAP completo, ou seja, "Banda larga" a 10 MHz ou "Banda Portadora" a 5 MHz em cabo coaxial.

À fim de conseguirmos tempos de resposta mais rápidos para mensagens, os nodos MAP/EPA e mini-MAP sacrificam algumas funcionalidades da arquitetura MAP completa. Desta forma, os seguintes serviços providos pelo MAP completo não são disponíveis:

- **Controle de fluxo completo** - As aplicações que usam os serviços EPA somente estarão habilitadas para manter uma mensagem pendente (não reconhecida) por vez, podendo, desta forma, ter uma taxa de saída reduzida;

- **Conceito de conexão** - Os serviços EPA são somente não orientados a associação;

- **Entrega da mensagem garantida com alta qualidade** - Os serviços EPA não garantem entrega da mensagem por não terem a camada de Transporte;

- **Entrega global de mensagens** - Serviços EPA não podem ser usados para entrega com reconhecimento de mensagens fora do segmento local;

- **Comprimento indefinido de mensagens** - O tamanho da mensagem é limitado quando se usa os serviços EPA ao tamanho máximo da Unidade de Dados do Protocolo da camada de Enlace (LPDU);

- **Serviços de Sessão** - Não pode haver ressincronização dos diálogos entre aplicações devido à ausência da camada de Sessão;

- **Serviços de Apresentação** - Devido a sua inexistência não é possível negociar ou trocar codificação de sintaxe abstrata durante uma Associação de Aplicação; a sintaxe de Apresentação deverá ser conhecida pelas aplicações;

Convém ressaltar que os serviços de Diretório mini-MAP e MAP são entidades separadas e distintas. Um Diretório mini-MAP proverá apenas um serviço de tradução nome-endereço para o usuário.

Há, também, uma compatibilidade entre a arquitetura de gerenciamento da rede usada para os nodos MAP/EPA e o gerenciamento usado para qualquer nodo MAP em geral.

3.3. Especificação de Mensagem da Manufatura (MMS)

3.3.1. Introdução

A Especificação de Mensagem da Manufatura (MMS), é um protocolo da camada de Aplicação proposto pela ISO com o nome ISO 2nd DP 9506 (MMS) que é idêntico ao "EIA Project 1393A Draft 6" (RS-511). A arquitetura MAP adotou-o na sua última versão (V3-0), que deverá ficar congelada por alguns anos.

Este protocolo tem por objetivo facilitar a integração em ambientes industriais de dispositivos programáveis, tais como Controladores Lógicos Programáveis (PLC´s), Controladores de Robôs (RC´s), Comandos Numéricos Computadorizados (CNC´s) e outros equipamentos de controle de processos, tais dispositivos podem ser de fabricantes diferentes e de complexidade diversa.

Os serviços fornecidos pelo MMS serão disponíveis ao usuário para que ele possa realizar aplicações distribuídas dentro do ambiente MAP.

3.3.2. Posicionamento do MMS em um Sistema de Produção

3.3.2.1. Introdução

À fim de posicionar a Especificação de Mensagem da Manufatura em um sistema de Produção e entender os diversos conceitos abstratos apresentados pelo mesmo, introduzimos nas seções seguintes uma visão geral dos aspectos de controle dos sistemas de manufatura reais. Os conceitos que não fazem parte da norma MMS são apresentados em tracejados nas figuras, destacando desta forma os pontos fundamentais para o entendimento e interpretação da mesma.

3.3.2.2. Sistema de Manufatura Real (RMS)

Por Sistema de Manufatura Real (RMS) entende-se todos os aspectos de um processo de manufatura ao qual alguma operação de controle está associada.

O Sistema de Manufatura Real (RMS) é constituído por diversos Dispositivos de Manufatura Real (RMD), fortemente acoplados; estes dispositivos são agrupados em unidades elementares (RMES) com acoplamento fraco entre si, comunicando-se geralmente através de serviços de mensagens. Como exemplo de Dispositivo de Manufatura Virtual (RMD) podemos citar: válvula motorizada, sensor de pressão, sensor de temperatura, reguladores, etc..

3.3.2.3. Sistema de Controle (CS)

A interconexão de um conjunto de equipamentos de processamento de dados e de um sistema de comunicação que assegura a operação das tarefas distribuídas de controle e monitoração do processo de produção é chamado de Sistema de Controle (CS) associado ao Sistema de Manufaturea Real (RMS).

O Sistema de Controle (CS) conhece somente uma imagem virtual do ambiente de produção. A funcionalidade de cada equipamento real é representada por um modelo abstrato, que especifica o seu comportamento observado externamente. Este modelo é chamado Dispositivo do Sistema de Controle (CSD), e possui uma imagem virtual de uma parte ou de todo Dispositivo de Manufatura Real (RMD), chamada de Dispositivo de Manufatura Virtual (VMD).

3.3.2.4. Dispositivo do Sistema de Controle Virtual (VCD)

O Dispositivo do Sistema de Controle Virtual (VCD) é uma unidade do Dispositivo do Sistema de Controle (CSD) constituído da associação de um Processo de Aplicação (AP) e de uma Entidade de Comunicação (CE) com capacidade de comunicar. A figura 11 relaciona os elementos apresentados.

Existem tantos VCD´s quanto AP´s dentro de um CSD. Somente uma Entidade de Comunicação (CE) pode existir em um Processo de Aplicação (AP) dentro de um Dispositivo de Sistema de Controle (CSD).

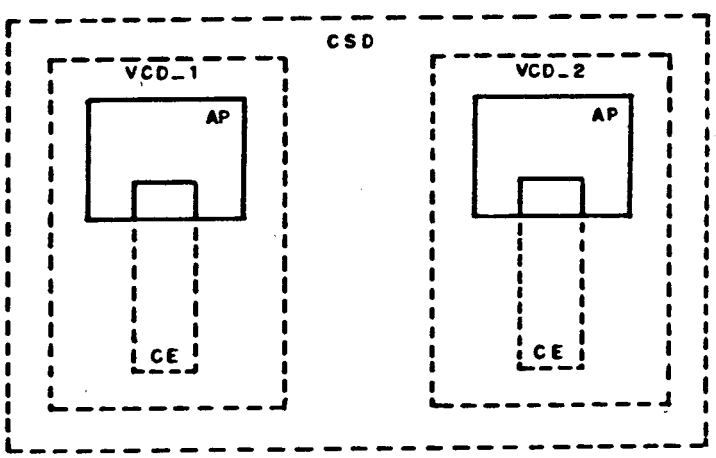


Fig. 11: Dispositivo do Sistema de Controle

3.3.2.5. Processo de Aplicação (AP)

Um Processo de Aplicação (AP) é uma representação abstrata responsável pela execução de uma ou mais tarefas de tratamento da informação, como parte da realização da aplicação distribuída, dentro de um sistema aberto real.

UM AP representa a participação de um sistema aberto real na tarefa distribuída de controle e monitoração do Sistema de Manufatura Real. Esta participação nas funções de processamento de informação distribuída exige a cooperação de dois ou mais Processos de Aplicação, através do compartilhamento de informações comuns. Por esta razão deve-se ter um modelo comum do ambiente de manufatura, que no caso particular do MMS é dado em termos de um conjunto de "objetos" abstratos.

As interações entre AP's modelizam a inter-operação de sistemas abertos reais, no ambiente OSI.

Um Processo de Aplicação (AP) pode reunir um conjunto de Dispositivos de Manufatura Virtual (VMD's) associados a uma mesma aplicação dentro de um Dispositivo do Sistema de Controle (CSD). Um CSD pode incluir diversos AP's.

A atividade de um Processo de Aplicação (AP) é suportada por uma Invocação do Processo de Aplicação, que descreve uma instância de um AP, para uma execução particular de um dado procedimento em um dado instante.

Na figura 12, é apresentado o modelo geral de um Sistema de Manufatura Real e do seu Controle.

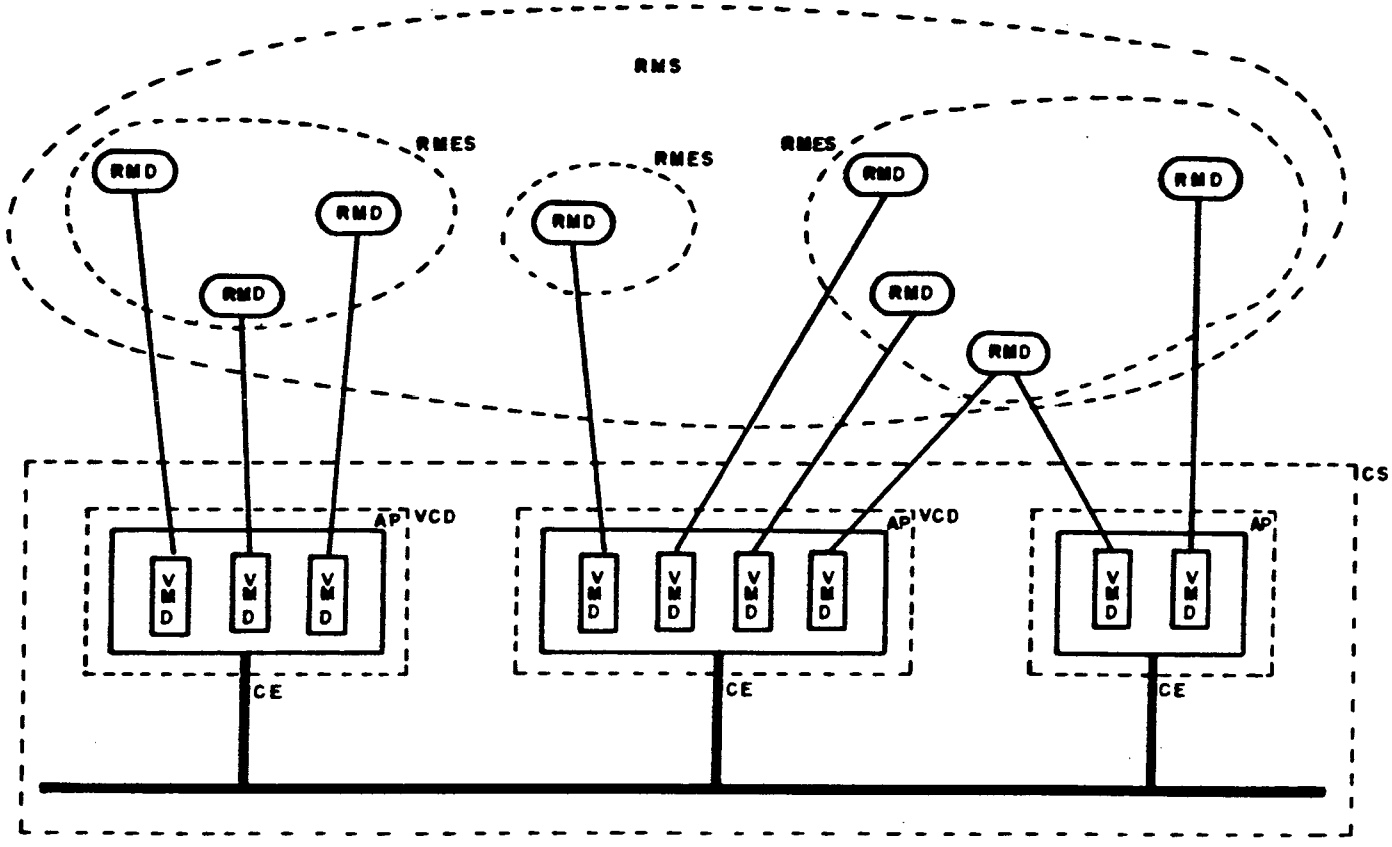


Fig. 12: Modelo Geral

3.3.2.6. Entidade de Comunicação (CE)

A Entidade de Comunicação (CE) é um elemento de um Dispositivo de Controle Virtual (VCD), apresentado na figura 13, que torna possível a comunicação com outro VCD que participa da realização de uma aplicação distribuída. As funções de comunicação são realizadas a partir dos Elementos do Serviço de Aplicação (ASE), e dos outros elementos de serviço das demais camadas.

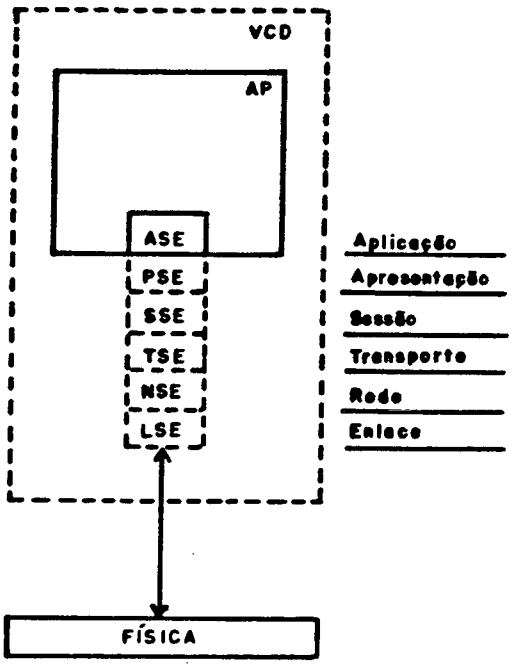


Fig. 13: Dispositivo de Controle Virtual

Podemos notar que os ASE´s formam a intersecção entre o Sistema de Controle e o Processo de Aplicação, que é a representação funcional do Sistema de Manufatura Real (RMS).

Os outros elementos do serviço das demais camadas são independentes do Sistema de Controle e são chamadas pelos ASE´s para a realização de suas funções de comunicação.

3.3.2.7. Entidade de Aplicação (AE)

A Entidade de Aplicação (AE) é um elemento ativo dentro de um Processo de Aplicação de um Sistema Aberto, e representa um conjunto de funções de comunicação associado a um VCD, de forma a permitir as interações de AP's envolvidas na comunicação OSI. Essas funções de comunicação são garantidas por um conjunto de elementos de serviço de aplicação (ASE's), apresentados na figura 14.

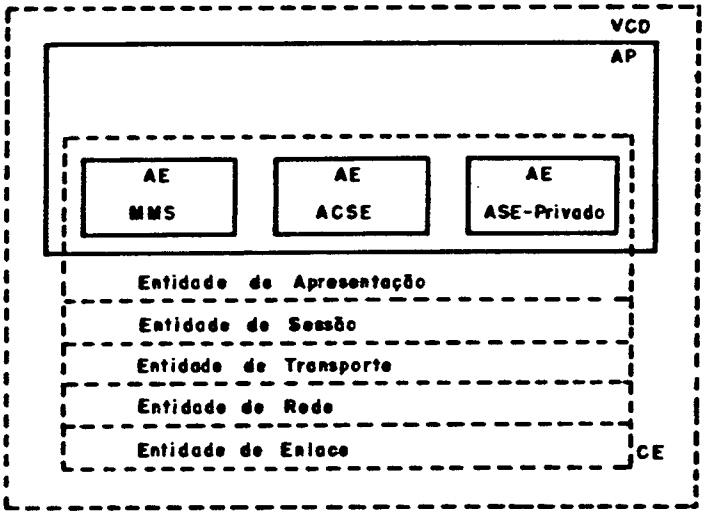


Fig. 14: Posicionamento da AE em um AP

Um ASE é um conjunto coerente de funções integradas que fornecem um conjunto de recursos de comunicação visando algum objetivo particular. Em especial, vamos nos interessar em descrever o MMS ("Message Manufacturing Specification"), que é

modelado como um ASE atualmente adotado para as comunicações com dispositivos de manufatura, nos sistemas de produção.

A relação de cooperação entre duas invocações de Entidades de Aplicação (AE) com o objetivo de trocar informação e coordenar sua operação conjunta é chamada Associação de Aplicação.

3.3.3. Técnica de Descrição do MMS

Todos os modelos dos dispositivos MMS e os procedimentos de serviços MMS são descritos através do uso da técnica de Modelo Objeto Abstrato.

Nesta técnica, cada objeto é uma entidade abstrata que representa suas características por atributos, que podem vir a ser modificados através da ação dos serviços MMS que atuam sobre estes objetos. Alguns dos atributos destes objetos são condicionais, dependendo de valores de outros atributos do mesmo objeto, associados numa expressão de restrição.

Na Implementação MMS, um sistema real mapeia os conceitos descritos no modelo para o dispositivo real. O MMS define um número de 15 objetos, que são: "Transaction", "Domain", "Program Invocation", "Unnamed Variable", "Named Variable", "Scattered Acess", "Named Variable List", "Named Type", "Semaphore", "Semaphore Entry", "Operator Entry Station", "Event Condition", "Event Action", "Event Enrollment" e "Journal".

O uso dos serviços MMS permite criar ou destruir objetos, por exemplo variáveis-MMS, condições de evento ("Event Condition"), domínios ("Domain"), etc., ou ainda trocar o estado desses objetos.

Em conseqüência, os objetos MMS, podem ser alocados de maneira estática ou dinâmica. No caso de uma alocação estática, o objeto não pode ser destruído e sua existência é pré-definida pelo servidor MMS; caso contrário, pode ser criado e destruído através das ações dos serviços MMS.

3.3.4. Descrição Conceitual do MMS

3.3.4.1. Apresentação Geral

MMS é considerado dentro da camada de Aplicação (ISO 7496), do Modelo de Referência OSI da ISO, como um Elemento de Serviço de Aplicação (ASE) e tem como objetivo facilitar a interconexão de sistemas de processamento de informação.

Os serviços MMS podem ser usados por qualquer outro elemento de serviço da camada de Aplicação ou por outros elementos do Processo de Aplicação (AP). Os serviços são providos através do protocolo MMS, o qual, faz uso de serviços disponíveis no Elemento de Serviço de Controle da Associação (ACSE, ISO 8649) e na camada de Aplicação (ISO 8822), para fornecer um conjunto de regras a serem observadas na comunicação entre entidades MMS homólogas.

3.3.4.2. Conceitos sobre Usuário do Serviço MMS

Usuário MMS: É visto como uma parte do Processo de Aplicação (AP) que utiliza os serviços oferecidos pela Especificação de Mensagem da Manufatura (MMS).

São adotados as seguintes convenções para definir os diversos tipos de usuário:

Em relação ao estabelecimento de contexto:

- **Usuário Chamador:** É o usuário MMS que toma a iniciativa de estabelecer o contexto MMS através da emissão da primitiva de Pedido do serviço "Initiate", permitindo assim o estabelecimento do diálogo entre usuários MMS;
- **Usuário Chamado:** é o usuário que responde ao pedido de estabelecimento do contexto MMS através da emissão de primitiva de Resposta do serviço "Initiate".

No Contexto MMS:

O contexto MMS fornece, uma vez estabelecido, uma especificação dos elementos do serviço MMS e as semânticas de comunicação a serem usados durante o tempo de vida de uma Associação de Aplicação.

Os usuários passam a ser considerados como:

- **Usuário requisitor:** É o usuário MMS que emite primitivas de Pedido de serviço. Este usuário pode ser considerado como:

    . **Emissor:** quando emite a primitiva de Pedido de Serviço;

    . **Receptor:** quando recebe a primitiva de Confirmação de serviço;

- **Usuário Respondedor:** É o usuário MMS que emite primitivas de Resposta de Serviço quando um determinado serviço foi solicitado.

    . **Emissor:** quando emite a primitiva de Resposta de serviço;

    . **Receptor:** quando recebe a primitiva de Indicação de serviço;

O princípio de base da comunicação no MMS é o modelo de interação cliente/servidor onde um lado do sistema (cliente) pede ao outro lado (servidor) alguns serviços. Em conseqüência, é entendido como:

- **Servidor de um serviço:** um usuário MMS que pode ter o comportamento externo de um Dispositivo de Manufatura Virtual (VMD) para uma instância particular de pedido de serviço.

- **Cliente de um serviço:** um usuário MMS que faz uso do VMD para alguma finalidade particular através da emissão de primitivas de Pedido de serviço.

3.3.4.3. Conceitos sobre Provedor do Serviço MMS

Provedor MMS: É visto como a parte da Entidade de Aplicação (AE) que provê os serviços MMS através da troca de PDU´s MMS. A distinção entre usuários MMS e provedor MMS é uma abstração, e pode não corresponder obrigatoriamente a realização de MMS em qualquer sistema.

A informação entre o usuário MMS e o provedor MMS é garantido através do uso das primitivas de serviço, através dos quais pode ser realizada a passagem de parâmetros.

O provedor MMS é entendido como pares de Máquinas de protocolo de Mensagem da Manufatura (MMPM) que enviam e/ou recebem PDU´s.

3.3.5. O Dispositivo de Manufatura Virtual (VMD)

3.3.5.1. Conceitos Gerais

O Dispositivo de Manufatura Virtual (VMD), do MMS, é a abstração que representa as potencialidades de um servidor real do ponto de vista dos recursos e das suas funcionalidades associadas. É na implementação que se fará o mapeamento de todos os objetos MMS do modelo VMD nas funcionalidades de um Dispositivo de Manufatura Real (RMD).

O VMD modela então o comportamento visível externamente do servidor MMS. Do ponto de vista da conceituação ISO, um VMD constitui uma parte do processamento da informação, que existe dentro de um Processo de Aplicação (AP) do tipo servidor MMS, tornando disponível um conjunto de recursos e funcionalidades para o cliente MMS.

Um Processo de Aplicação poderá ter nenhum ou diversos VMD´s, sendo que cada um representa um Dispositivo de Manufatura Real dentro do Processo de Aplicação. Um VMD pode conter exclusivamente nenhuma ou mais Entidades de Aplicação (AE). Cada AE representa um conjunto de características de comunicação usado pelo VMD.

Um dado VMD é endereçado por um ou mais AE´s, que por sua vez está associada, cada uma, a um simples Ponto de Acesso ao Serviço de Apresentação (PSAP).

Na figura 15, é apresentado o Processo de Aplicação do tipo servidor MMS;

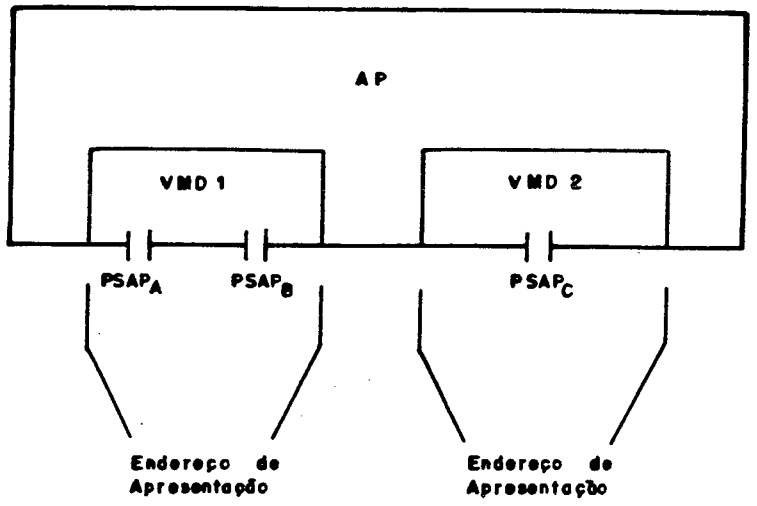


Fig. 15: Processo de Aplicação Servidor MMS

A operação de um VMD é descrita através de objetos abstratos que são manipulados por ele, e do conjunto de operações que podem ser realizados nesses objetos através do uso do servidor MMS.

3.3.5.2. Estrutura de um VMD

Um VMD sempre tem no mínimo um ou mais Domínios, nenhuma ou mais Estação Operador, um Arquivo Virtual opcional e uma Função Executiva. A figura 16 apresenta a estrutura do VMD.

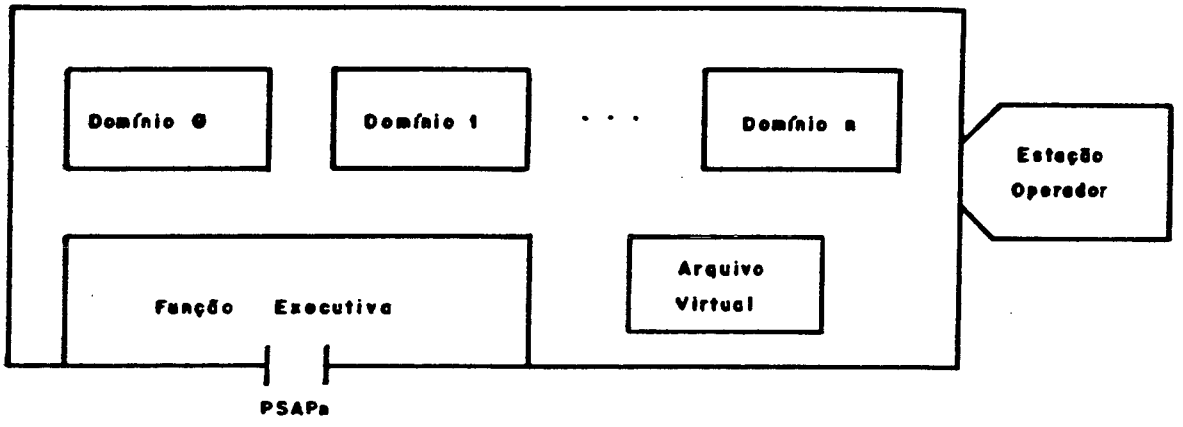


Fig. 16: Estrutura de um VMD

Função Executiva:

É um elemento primordial de um VMD, pois permite o gerenciamento do acesso aos recursos primitivos de um VMD, como por exemplo, memória, processador, portas de E/S, etc. Inclui todos os aspectos de uma operação local do VMD que são visíveis aos seus clientes.

Domínio:

Representa um subconjunto dos recursos do VMD que são usados para uma finalidade específica, portanto, um VMD pode ser composto de diversos Domínios, um para cada especialidade. A alocação das potencialidades de um VMD a um Domínio podem ser feitas de forma estática ou dinâmica, como qualquer objeto.

Um Domínio é identificado por um único nome no qual Objetos MMS são identificáveis. A maior parte dos objetos MMS são sempre associados a um domínio, sendo que os objetos Domínios contém principalmente programas e dados. Associados com este objeto estão seu nome, uma lista dos recursos do VMD, que foram pedidos emprestados ao VMD, e uma lista de objetos subordinados (como por exemplo: variáveis, eventos, etc, possuindo um tempo de vida dependente do domínio).

O atributo "Domain Content" permite a identificação da informação contida no próprio Domínio. A interpretação de Domínios é feita através do seu uso pela "Invocação do Programa", que será destacado à seguir.

Invocação do Programa:

É uma abstração do que chamamos normalmente de programa /Rhein 86/. Invocação de Programa consiste de um conjunto de procedimentos e elementos de dados contidos dentro de Domínios.

É identificado por um nome, que deve ser único para todas as Invocações de um determinado VMD, e os atributos de uma

Invocação de Programa são os seguintes: nome, estado, indicação da possibilidade de apagar e lista de Domínios que são incluídas na sua definição. Deve existir no mínimo, um objeto Domínio com o qual o objeto de Invocação de Programa se liga; neste caso o conteúdo do domínio normalmente serve ou como um algorítmo para a Invocação do Programa ou como dado nos quais tais algorítmos trabalham, ou uma combinação de ambos.

Arquivo Virtual:

Um arquivo virtual consiste de um conjunto de arquivos nomeados, que serve para armazenamento de dados e programas. Esses arquivos podem ser usados por alguns serviços do Domínio e Gerenciamento do Programa. Um arquivo virtual é um elemento opcional dentro de um VMD e no seu relacionamento com o ambiente OSI, é visto como um subconjunto do arquivo virtual do FTAM.

Estação Operador:

É uma classe de dispositivos que não são visíveis ao cliente MMS como recursos de VMD específicos atribuídas no Domínio. É chamado por fora como objeto separado. Uma VMD pode ter nenhuma ou diversas Estação Operador.

Objeto Transação:

Um objeto transação é criado quando o VMD recebe uma primitiva de Indicação de serviço para algum serviço MMS confirmado; este objeto gerencia o processamento deste serviço particular e é destruído após o usuário MMS emitir uma primitiva de resposta de serviço para esta instância de serviço.

3.3.6. Serviços MMS

3.3.6.1. Apresentação Geral

Os protocolos de comunicação, incluindo o MMS, possuem dois níveis de descrição, como visto no capítulo anterior. A primeira é a descrição do serviço que especifica o comportamento global do sistema, e a segunda é a descrição do protocolo que especifica o comportamento em termos de entidades cooperantes, com o fim de realizar o serviço.

Os serviços MMS são modelos abstratos que descrevem: as interações entre usuários do serviço; as ações e eventos primitivos do serviço com os dados associados; as relações e seqüências válidas destes; os requisitos procedurais associados com a execução do pedido de serviço.

Por outro lado, o protocolo MMS fornece um conjunto de procedimentos que permitem a transferência de informações de dados e controles de Entidades de Aplicação (AE), dentro do contexto MMS. Esses procedimentos correspondem a interações entre Entidades de Comunicação (CE) tais como:

- trocas de PDU entre Entidades de Aplicação (AE) homólogas;
- trocas de primitivas de serviço MMS, no limite entre usuário MMS e provedor MMS;
- trocas de primitivas de serviço ACS, entre o provedor MMS e o ACSE;e
- trocas de primitivas de serviço da camada de Apresentação, entre o provedor MMS e o elemento de serviço de Apresentação.

A estruturação imposta ao protocolo permite grande variação e opções dos serviços disponíveis,de forma que tenhamos uma grande variedade de aplicações. Assim, a implementação mínima não precisa conter todos os possíveis serviços, somente um conjunto mínimo necessário para uma determinada classe de aplicação. O acesso a estes serviços é feito através da troca de primitivas de serviço do Modelo de Referência OSI (Pedido, Indicação, Resposta e Confirmação).

Cada serviço é associado a pelo menos um Bloco de Construção de Conformidade (CBB). Um CBB indica o conjunto de serviços que são dependentes de outros serviços pertencentes ao mesmo CBB, ou seja, todos os serviços que pertençam a um CBB devem ser implementados, para que cada serviço possa ser executado plenamente (um depende do outro).

Os serviços suportados pela Especificação de Mensagem da Manufatura são agrupados em 10 classes principais que serão descritas resumidamente nas próximas seções e apresentadas em anexo.

3.3.6.2. Serviços de Gerenciamento de Contexto

Os serviços de Gerenciamento de Contexto possibilitam uma negociação entre dois sistemas que desejam comunicar-se à fim de estabelecer os recursos necessários para que o diálogo entre dois usuários, através do serviço MMS, seja realizado.

Como MMS é um protocolo orientado a associação, os serviços de Gerenciamento de Contexto são serviços básicos sempre utilizados para a iniciação, cancelamento, rejeição e conclusão da transação entre programas de aplicação cooperantes. Estes serviços fazem uso dos serviços fornecidos pelo ACSE para gerenciar uma Associação de Aplicação.

Iniciação da comunicação ("Initiate"), conclusão da comunicação de forma negociada ("Conclude"), conclusão da comunicação de forma abrupta e sem negociação ("Abort"), Cancelamento de pedidos de serviços pendentes ("Cancel"), e notificação de erros pelo provedor ("Reject") são os serviços disponíveis desta classe.

3.3.6.3. Serviços de Suporte do VMD

Os serviços de suporte do VMD tornam possível ao usuário MMS controlar o modelo abstrato representativo do Sistema de Manufatura Real, permitindo:

- Conhecer o estado lógico de um VMD, ("**status**"); receber uma mensagem não solicitada sobre o estado do VMD ("**UnsolicitedStatus**");

- Conhecer lista (ou parte da lista) dos vários objetos definidos, ("**GetNameList**");

- Identificar as informações específicas do equipamento ("**identify**") (marca, modelo, revisão) na Entidade de Aplicação MMS;

- Substituir um nome de objeto ("**Rename**")

3.3.6.4. Serviços de Gerenciamento de Domínio

Domínios são elementos abstratos pertencentes ao modelo MMS do Dispositivo de Manufatura Virtual (VMD), como vistos na secção anterior. Os domínios podem ser dinâmicos, sendo criados e removidos, localmente ou através dos serviços MMS.

Os Domínios são manipulados pelos usuários clientes através de um conjunto de serviços definidos no servidor MMS e que realizam operações tais como: transferência de Domínio do cliente ao servidor ("DownLoad"), a obtenção de Domínio do servidor pelo cliente ("UpLoad"), carga pelo servidor de um Domínio, a partir de um arquivo, salvamento pelo servidor de um Domínio num arquivo, a destruição de um Domínio pelo servidor e a obtenção pelo cliente dos atributos de um Domínio.

3.3.6.5. Serviços de Gerenciamento de Invocação de Programa

Invocação de Programa é um elemento abstrato pertencente ao modelo MMS do Dispositivo de Manufatura Virtual (VMD). As Invocações de Programas podem ser dinâmicas em natureza, surgindo e sendo criados e removidos localmente ou através de serviços MMS.

Estes serviços permitem ao usuário cliente MMS, criar e destruir o objeto Invocação de Programa, iniciar, parar, reassumir, reiniciar este objeto e determinar os atributos deste objeto.

3.3.6.6. Serviços de Acesso a Variável

São serviços que permitem ao usuário MMS cliente acessar variáveis através de operações de leitura/escrita de valores variáveis armazenadas no Dispositivo de Manufatura Virtual, além de definir nomes e tipos de variáveis.

3.3.6.7. Serviços de Gerenciamento de Semáforo

São serviços que permitem a sincronização, controle e coordenação de recursos compartilhados entre usuários MMS, através da definição e o controle de dois tipos de semáforos. O semáforo "Token" permite simples ou multiplos proprietários; o semáforo "Pool" que permite uma alocação dinâmica ou explícita de "Tokens" nomeados.

3.3.6.8. Serviços de Comunicação com o Operador

Este serviços fornecem um mecanismo para comunicação com a estação operador de um dispositivo remoto, fornecendo instruções e/ou lendo as entradas do mesmo, através de dois serviços básicos: "Input" que recebe dados da estação operador e "Output" que envia uma mansagem a uma estação operador.

3.3.6.9. Serviços de Gerenciamento de Eventos

São serviços que permitem ao usuário MMS cliente definir e gerenciar eventos em um VMD e obter notificações de ocorrência de eventos específicos em situações de tempo real, permitindo desta forma a sincronização de ações.

O MMS define três objetos adicionais no VMD para modelar os aspectos específicos do gerenciamento dos eventos MMS:

- objeto "Condição de Evento" relacionados com deteção e priorização de eventos;
- objeto "Ação de Evento" relacionados com a execução de serviços MMS sob a ocorrência de um evento; e
- objeto "Registro de Evento" que une uma dada "Condição de Evento" a uma dada "Ação de Evento".

3.3.6.10. Serviços de Gerenciamento de Ocorrências (Jornal)

Estes seviços permitem a criação de arquivos com o registro e armazenamento de ocorrências de eventos e/ou variáveis de interesse em conjunção com eventos, associando a data, o número da ocorrência e o tempo em uma lista, para manter uma história dos principais eventos e dados ocorridos no sistema.

3.3.6.11. **Serviços de Gerenciamento de Arquivos** (Anexo)

Estes serviços permitem a abertura ("FileOpen"), transferência ("FileRead") e fechamento ("FileClose") do arquivo virtual definido no servidor MMS, além de delir ("FileDelete"), trocar o nome ("FileRename") e obter atributos de grupos de arquivos ("FileDirectory").

3.4. Exemplo de aplicação : A Célula Flexível de Usinagem (CFU)

Para esclarecer os conceitos envolvidos nos modelos utilizados pelo MMS, será feita a descrição de uma célula flexível de usinagem, mostrando a sua representação e programação.

3.4.1. Apresentação Geral

3.4.1.1. A Usinagem

A usinagem é uma das etapas envolvidas no processo de fabricação industrial e consiste na modificação planejada de uma peça através do uso de algumas ferramentas que realizam diversas operações como cortar, furar, etc.

3.4.1.2. Elementos da CFU

- Uma esteira para transportar os vários tipos de peças disponíveis;
- Dois sensores P1 e P2 para detectar a presença de peças sobre a esteira;
- Uma câmera para identificar e classificar as peças;

- Uma máquina ferramenta para fazer a usinagem de cada peça;
- Um robô (Rp) para retirar a peça da esteira e colocá-la na máquina ferramenta;
- Um robô (Ra) para retirar a peça usinada e colocá-la no armazém apropriado;
- Três armazéns para depositar cada tipo de peça usinada;
- Um lixeiro para depositar as peças rejeitadas.

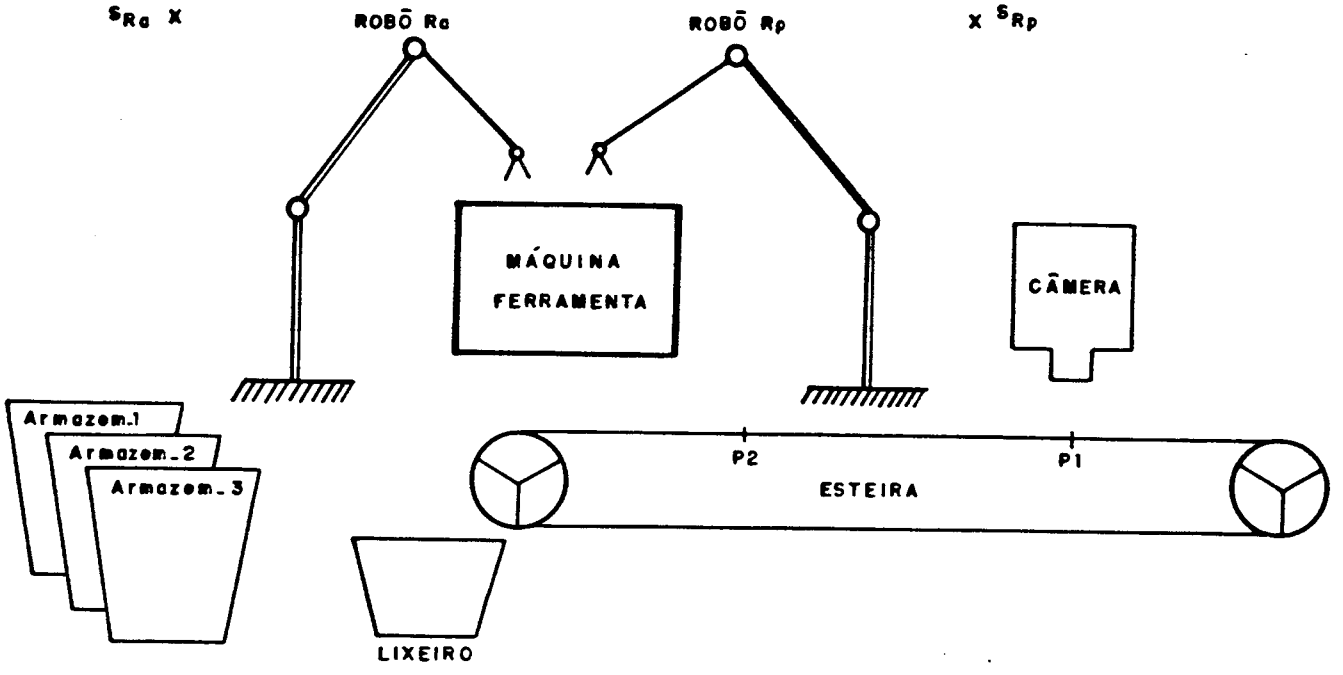


Fig. 17: A Célula Flexível de Usinagem

3.4.1.3. Descrição da CFU (figura 17)

Peças de diversos tipos são **colocadas** na esteira para serem **transportadas.** Ela permanece em movimento até que o sensor P1 e/ou sensor P2 **detecte** uma peça, quando então ela ficará parada aguardando uma ordem de movimento.

Quando a peça está em P1, a câmera fará a **visualização** para a posterior **identificação** da peça, permitindo desta forma, classificá-la em função de modelos pré-estabelecidos para ser usinada futuramente. No final desta operação, serão **armazenadas** as informações obtidas num arquivo a ser fornecido ao robô Rp e a máquina ferramenta, além de ser dado um aviso para o reinício do funcionamento da esteira.

Quando a peça está em P2 e tiver sido classificada como um tipo válido, o robô Rp é posto em movimento a partir de sua posição de repouso Srp até a posição P2 onde ele **apanha** a peça, juntamente com as informações que a identificam, liberando a esteira.

O robô Rp aguarda até que a Máquina Ferramenta fique desocupada e que a área ao redor da mesma fique liberada pelo robô Ra, para **colocar** a peça para usinar. O espaço ao redor do torno é então liberado pelo robô Rp que retorna para a sua posição de repouso Srp, assinalando que a usinagem pode começar e que o robô Rp está disponível para nova operação.

A máquina_ferramenta então **carrega** a peça para usinar e executa uma série de operações como **resfriar peça, operar ferramenta_A, operar ferramenta_B, operar ferramenta_C, lavar peça** com uma seqüência de operação que é função do tipo da peça identificada. Somente uma única ferramenta pode ser operada por vez, e para operar cada ferramenta é necessário primeiro **pegar** a ferramenta escolhida, e após o uso **liberá-la.** Após o processo de usinagem a peça será **descarregada.**

A partir da notificação deste evento, o robô Ra sai de sua posição de repouso Sra, se o espaço ao redor não estiver sendo ocupado pelo robô Rp, e **retira** a peça do torno, liberando desta forma o espaço ao redor do mesmo; a informação do tipo de peça usinada é também fornecido ao robô Ra.

A peça será então **armazenada** no armazém correspondente ao tipo de peça usinada, que avisa ao robô Ra quando ele pode ou não armazenar peças. O armazém é substituído quando está cheio.

Quando a peça está em P2 e tiver sido rejeitada, o robô Rp não é acionado e libera imediatamente o movimento da esteira, sendo a peça depositada no lixeiro.

3.4.1.4. As peças

As peças são os elementos que sofrerão o processo de usinagem. Elas são identificadas pelo seu tipo: tipo_1, tipo_2, tipo_3 ou rejeitada.

3.4.1.5. As Operações

São responsáveis pelas transformações sofridas pelas peças do ponto de vista de sua identidade dentro de uma célula de usinagem.

As operações possíveis são: alimentar, transportar, visualizar, identificar, apanhar peça, usinar, colocar peça, descarregar, retirar peça, armazenar, trocar armazém.

3.4.2. Conceitos para Descrição Funcional

A descrição funcional da célula flexível será facilitada a partir da utilização dos seguintes conceitos /Mazzola 88, Chochon 86/:

Atores:

Ator é todo elemento que interage com as peças e podem ser classificados em:

- **Executores:** realizam as tarefas de movimento e deslocamento de peças:
  - . Braço do robô Rp;
  - . Braço do robô Ra;
  - . Articulação do robô Rp;
  - . Articulação do robô Ra;
  - . Esteira;

- **Suporte:** mantém as peças em uma posição estável, podendo ser passivos ou ativos.

    . Garras do robô Rp;

    . Garras do robô Ra;

- **Sensores:** fornecem informações sobre o estado da célula.

    . Câmera;

    . Indicador de posição P1;

    . Indicador de posição P2;

- **Máquinas especializadas:** realizam as operações físicas específicas.

    . Máquina ferramenta;

- **Sistema de alimentação e descarregamento:** realizam as entradas e saídas de peças da célula.

    . Armazém_1;

    . Armazém_2;

    . Armazém_3;

    . Lixeiro.

Agentes:

Um agente agrupa um conjunto de atores que são necessários para a realização de uma tarefa:

- **Robô Rp:** braço (executor), articulação (executor), garra (suporte);

- **Robô Ra:** braço (executor), articulação (executor), garra (suporte);
- **Esteira:** esteira (executor), sensor P1 (sensor), sensor P2 (sensor);
- **Câmera:** câmera (sensor);
- **Máquina:** ferramenta_A, ferramenta_B, ferramenta_C, (máquinas especializadas).
- **Armazém:** Armazém_1, Armazém_2, Armazém_3 (sistema de descarregamento)

3.4.3. Programação da CFU

O sistema de programação da CFU está ligado aos aspectos de gerenciamento e controle da execução de uma tarefa de usinagem, que são estabelecidos através de um Modelo de Execução, hierarquicamente estruturado de acordo com a figura 18 em dois módulos componentes:

Sistema de Supervisão

Este sistema recebe do nível superior, obedecendo a determinados critérios de planejamento, o Plano da Tarefa de usinagem, que determina todas as evoluções possíveis das peças dentro da célula, através de um conjunto de ações elementares a serem realizadas pelos agentes. A execução dessas ações elementares modifica o Estado Espacial da célula, que representa a localização das peças, dos sítios (agrupamentos localizados de elementos que tem a mesma função) e dos executores dentro da célula.

O Sistema de Supervisão retorna para o nível superior a informação do Estado da Tarefa, o qual indica a localização e a situação das peças dentro da célula.

Para a estruturação deste módulo, deve-se considerar: os aspectos funcionais, utilizando os conceitos de ator e agente da CFU; os aspectos do gerenciamento do paralelismo e da sincronização através da sinalização entre ações elementares executadas pelos agentes; os aspectos do gerenciamento do acesso correspondente aos recursos da célula, representados pelos atores, para a execução de cada ação elementar.

As ações elementares a serem executadas por cada agente, são traduzidas em um conjunto de Primitivas Funcionais, que são enviadas ao **Sistema de Comando** para ativar cada ator que constitui o agente considerado. O Sistema de Supervisão recebe, então, uma Resposta indicando o sucesso na execução da Primitiva e possivelmente um resultado.

Sistema de Comando

Faz a interface com o sistema de manufatura real, traduzindo as Primitivas Funcionais, recebidas do sistema de supervisão, em pedidos que serão enviados ao sistema de controle de cada componente da célula. Este nível é responsável pelas funções dos atores disponíveis na célula, cujos estados correspondem ao Estado Funcional da célula.

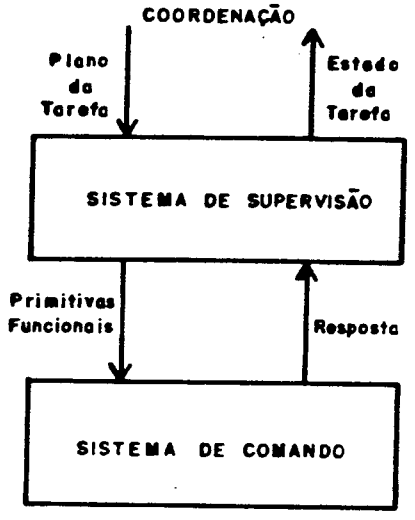


Fig. 18: Modelo de Execução

3.4.4. Posicionamento dos Serviços MMS na CFU

O exemplo e os conceitos apresentados tem o propósito de posicionar os serviços MMS em um sistema de produção industrial, relacionando-os com a descrição da Célula Flexível de Usinagem. Os elementos da CFU que tem a mesma função e acessam ao mesmo espaço comum dentro da célula são agrupados em sítios, onde é possível manipular uma determinada peça.

Uma possível implementação deste exemplo, pode ser feita sobre uma rede MAP banda portadora, a nível da célula, onde cada componente do Sistema de Comando terá, por exemplo, uma arquitetura mini-MAP. O Supervisor da Célula, recebe as ordens (plano da tarefa) do Coordenador da Usina, e comanda em tempo real as tarefas de usinagem das peças, transmitindo estas ordens aos diferentes robôs e equipamentos de usinagem. Para permitir a comunicação com outras células, ou com os níveis superiores, e ao mesmo tempo coordenar e sincronizar as atividades da CFU, o sistema de supervisão deverá possuir uma arquitetura MAP/EPA.

3.4.4.1. O Sistema de Programação Geral

Cada sítio é modelado por um único Processo de Aplicação, que comunica-se com outro sítio através dos serviços oferecidos pelo MMS. Cada agente é representado por um VMD, dentro do sítio, fazendo o Sistema de Supervisão ver cada componente do Sistema de Comando como um servidor MMS, e ser visto como um cliente MMS. Cada ator que compõe um agente é associado a um Domínio do VMD, o qual agrupa as Primitivas Funcionais e os recursos necessários a sua execução.

A comunicação do Supervisor da célula com cada servidor é constituída por uma série de pedidos, geralmente seguidos de resposta, que podem atuar sobre variáveis MMS ou ativar um dado procedimento (por exemplo, pegar uma peça em uma posição e colocá-la em outra).

A CFU tem como característica principal um alto grau de flexibilidade, proporcionado através da configuração dinâmica resultante da aplicação dos serviços MMS, os quais realizam a interface entre o Sistema de Supervisão e as diferentes máquinas que formam o Sistema de Comando, tornando possíveis a execução e a sinalização de fim de execução das Primitivas Funcionais, independente do equipamento. Cada Primitiva Funcional corresponde a ativação de uma Invocação de Programa associado ao Domínio que representa o ator responsável pela sua execução.

Assim, o Modelo de Programação (figura 19) é constituído pelos agentes da CFU modelados como VMD's e tendo os domínios associados aos respectivos agentes, apresentados a seguir:

VMD_Rp : braço_Rp, articulação_Rp e garra_Rp (domínios);

VMD_Ra : braço_Ra, articulação_Ra e garra_Ra (domínios);

VMD_Est : esteira, sensor_P1 e sensor_P2 (domínios);

VMD_Cam : câmera (domínio);

VMD_Maq : ferramenta_A, ferramenta_B e ferramenta_C (domínios);

VMD_arm : armazém_1, armazém_2 e armazém_3 (domínios).

Associado a cada um dos robôs (VMD_Ra e VMD_Rp) temos um semáforo do tipo "Token" (Sem_Ra e Sem_Rp, respectivamente), para impedir que mais de uma peça seja manuseada simultaneamente por eles.

Para permitir a exclusão mútua entre os robôs Ra e Rp, no acesso à Máquina Ferramenta, associamos um semáforo do tipo "Token" (Sem_Maq) ao VMD_Maq; somente conseguirá acessar a máquina ferramenta quem detiver o controle deste semáforo.

Será utilizado o semáforo do tipo "pool" (com duas fichas nomeadas: Ficha P1 e Ficha P2) para controlar o movimento da esteira; ela somente poderá ser colocada em funcionamento quando nao tiver peça sendo visualizada (Ficha P1), ou em processo de ser apanhada pelo robô Rp (Ficha P2).

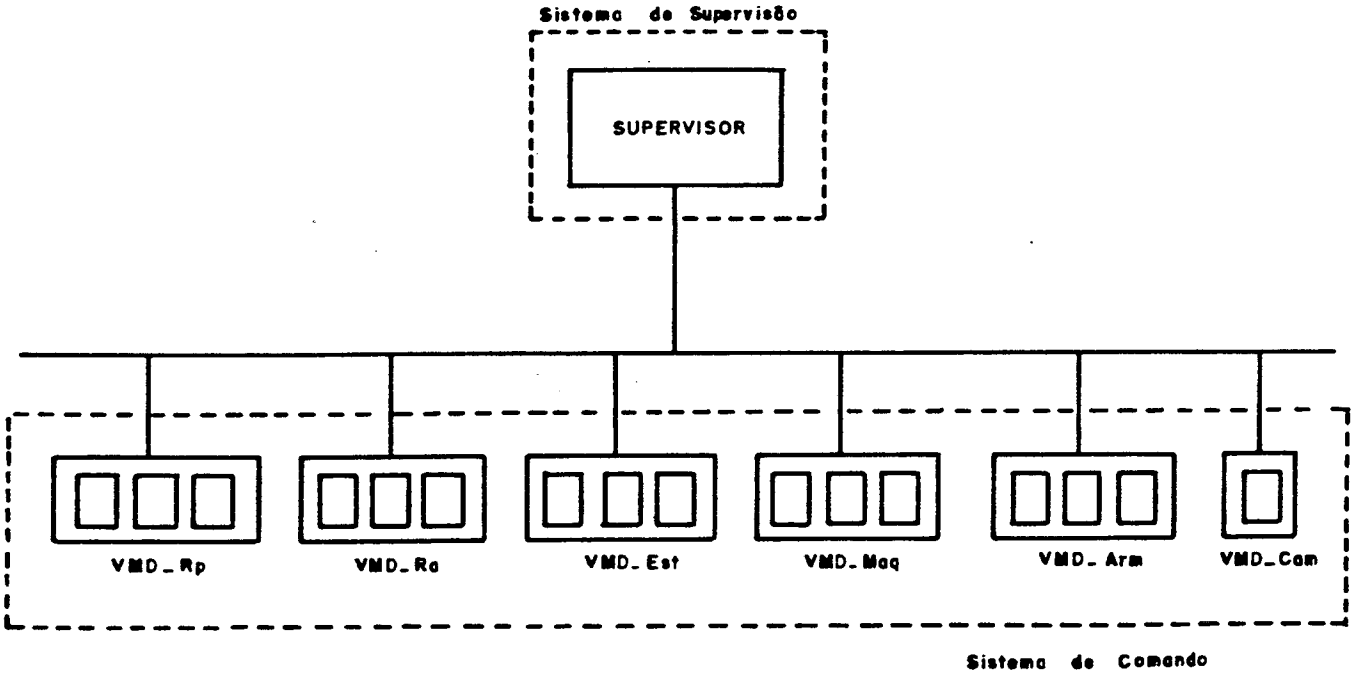


Fig. 19: Modelo de Programação CFU

Será também definido um objeto Condição de Evento associado a detecção de peças em P1 (Event_P1) e em P2 (Event_P2), de forma a coordenar o movimento da esteira e sincronizar as outras atividades relacionadas a este evento, como visualização pela câmara ou, carregamento pelo robô Rp, respectivamente.

3.4.4.2. Programação da Fase de Estabelecimento dos Recusos da Célula

Para permitir a Operação da Célula Flexível de Usinagem, o Supervisor da Célula inicialmente estabelece a comunicação com cada um dos equipamentos que constituem o sistema de Comando. Esta comunicação é estabelecida e mantida através dos Serviços de Gerenciamento de Contexto, constituído pelos serviços "Initiate", "Conclude", "Abort", "Cancel" e "Reject".

O contexto para comunicação é estabelecido fazendo uso do serviço "Initiate", o qual define além dos recursos necessários para a comunicação propriamente dita, quais serão os serviços MMS que deverão ser suportados, em cada equipamento. A figura 20 mostra o uso deste serviço e de alguns outros serviços usados na fase de troca de informações, necessários para preparar a operação da CFU; nesta figura não são apresentadas as primitivas de serviço necessárias para a realização desses serviços, ficando-as, entretanto, implícitas.

Para o estabelecimento dos recursos da célula, o supervisor inicialmente fará a identificação de cada equipamento ("Identify"), a realização de um diagnóstico desses equipamentos ("Status") e a obtenção do nome dos vários objetos MMS existentes nos mesmos ("GetNameList"). Além desses Serviços de Suporte do VMD, será utilizado o serviço "Rename", que permite redefinir o nome dos objetos MMS, necessário para a reconfiguração do sistema, e o serviço "UnsolicitedStatus", que permite obter o estado lógico de VMD, de forma não solicitada.

Serviços de Invocação de Programas são necessários para a realização das Primitivas Funcionais, que comandarão os atores da CFU. Constam dos serviços "CreateProgramInvocation", "DeleteProgramInvocation" e do serviço "Start" que durante a operação da célula possibilitará a execução de uma Invocação de Programa.

É nesta etapa que o supervisor define e cria todos os objetos não existentes que serão manipulados (e até destruídos) durante a operação da célula. Os objetos manipulados são a Invocação de Programa, Acesso a Variável, Semáforo e condição de Eventos, criados a partir dos serviços de gerenciamento correspondente. Além disso, são estabelecidos os valores iniciais para as variáveis MMS utilizadas no nível da célula e ainda são colocados os equipamentos em sua posição de operação.

Os Serviços de Gerenciamento de Domínio serão utilizados para carregar e trocar o conteúdo de domínios que não são estáticos, sendo um suporte para a flexibilidade computacional da CFU. Os serviços oferecidos são: "InitiateDownLoadSequence", "DownLoadSegment", "TerminateDownLoadSequence" e "DeleteDomain".

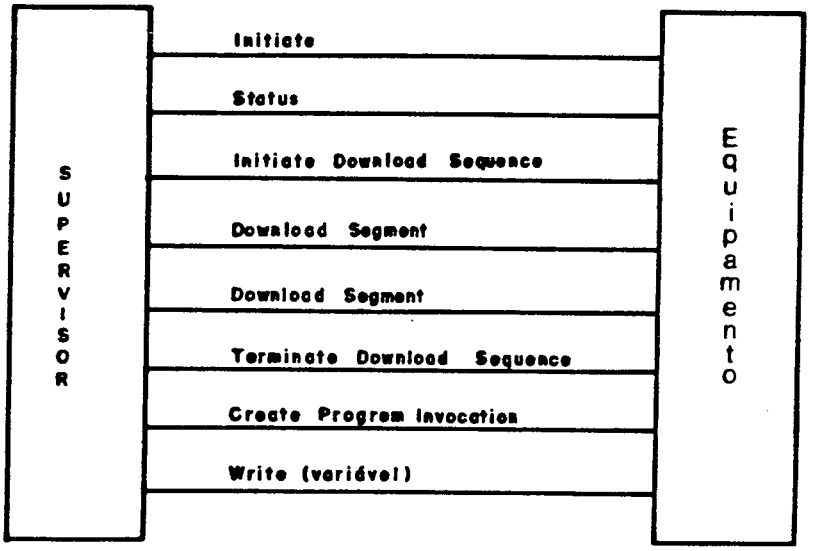


Fig. 20: Serviços MMS

3.4.4.3. Considerações Gerais

Associado com cada operação dos robôs Ra e Rp temos um semáforo que possibilita o controle de operação dos mesmos (Sem_Ra e Sem_Rp, respectivamente). Para isso, toda a Invocação de Programa correspondente a execução de Primitivas Funcionais, através do serviço "Start", será precedida pela tomada de controle do semáforo, através do serviço "Take Control"; após a confirmação de execução destas Primitivas os semáforos serão

liberados, pelo serviço "Relinquished Control". Da mesma forma os robôs Ra e Rp precisam possuir o controle do semáforo da máquina ferramenta (Sem_Maq) para poder colocar e retirar peças da mesma, impedindo o acesso simultâneo dos mesmos, além da operação de uma única peça.

O controle do movimento da esteira é feito com o uso de um semaforo com duas fichas nomeadas (ficha_P1 e ficha_P2); a esteira somente poderá ser colocada em movimento quando as duas fichas estiverem disponíveis. Para isso, associamos ao pedido do serviço para colocar a esteira em movimento o modificador "AttachToSemaphore", que é uma pré-condição para execução do serviço a ele associado, o qual ocorrerá quando o mesmo detiver as duas fichas. Será utilizado o modificador "AttachToEvent Condition" para atrasar a execução do serviço de parar a esteira, até a ocorrência de uma condição de evento especificada, que neste caso, é a detecção da presença de peças em P1 (Event_P1) ou em P2 (Event_P2). Este modificador estará associado a execução da Invocação de Programa responsável em parar a Esteira (serviço "Start" modificado), com peças em P1 e em P2.

3.4.4.4. Programação da Fase de Operação da Célula

Após todos os recursos serem estabelecidos, a CFU está pronta para operar, a partir das ativações das Invocações de Programa pelo Supervisor, baseado no plano da tarefa. Assim, a primeira Invocação de Programa a ativar corresponde a Primitiva Funcional para movimentar a esteira (AttachToSemaphore(ficha_P1,

ficha_P2).Start(Ligar)); a Primitiva seguinte é responsável em parar a esteira, na condição de existência de peça em P1 (AttachToEventCondition(Event_P1).Start(Parar)); outra Primitiva usada serve para parar a esteira, na condição de existência de peça em P2 (AttachToEventCondition(Event_P2). Start(Parar)).

A cada confirmação de serviço de parar a esteira, com a condição de existência de peça em P1, faz o supervisor realizar as seguintes operaçoes:

- Assumir o semáforo de controle da esteira (TakeControl (ficha_P1)) (impedindo que a mesma possa mover-se antes de terminar o processo de identificação);
- Ativar a visualização/identificação na câmera (Start (Visualizar));
- Ler a variável correspondente ao tipo da peça analisada (Read(Parâmetros_peça)) pela câmera;
- Liberar o semáforo de controle da esteira (Relinquish Control(ficha_P1)); e
- Colocar a esteira em funcionamento (AttachToSemaphore (ficha_P1,ficha_P2).Start(Ligar)) (que ocorrerá de forma efetiva quando os semáforos que controlam o movimento da esteira estiverem disponíveis, devido ao uso do modificador).

Por outro lado, a cada confirmação do serviço de parar a esteira, devido a presença de peça em P2, faz o Supervisor realizar as seguintes operações:

. Assumir o semáforo de controle da esteira (TakeControl (ficha_ P2));

. Assumir o semáforo de controle do robô Rp (TakeControl (Sem_Rp));

. Comandar o robô Rp para apanhar a peça (Start (Apanhar));

. Liberar o semáforo de controle da esteira (Relinquish Control(ficha_P2));

. Colocar a esteira em funcionamento (AttachToSemaphore (ficha_P1,ficha_P2).Start(Ligar));

. Assumir o semáforo de controle de acesso a máquina ferramenta (TakeControl(Sem_Maq));

. Colocar a peça para usinar (Start(Colocar));

. Liberar o semáforo de controle do robô Rp (Relinquish Control(Sem_Rp));

. Retornar o robô Rp para a posição de repouso (Start (Repousar));

. Atualizar as informações da peça a ser usinada (Write (Parâmetros_peça));

. Ativar as Invocações de Programas responsáveis pelas Primitivas Funcionais, necessárias para a realização da tarefa de usinagem, de acordo com o tipo da peça identificada ("seqüências de Start");

. Assumir o semáforo de controle do robô Ra;

. Comandar o robô Ra para retirar a peça (Start(Retirar));

. Liberar o semáforo de controle da máquina ferramenta (RelinquishControl(Sem_Maq));

. Colocar a peça no depósito correspondente (Start (Armazenar)), e

. Liberar o semáforo de controle do robô Ra.

Estas ações continuam até que um novo plano de tarefa seja recebido, quando então serão feitas as novas ações elementares, em função deste plano.

3.5. Conclusão

Neste capítulo foram apresentadas as diversas arquiteturas de comunicação para ambientes industriais adotadas no projeto MAP; este projeto foi baseado no Modelo de Referência da ISO e está sendo adotado como padrão internacional para interligar os diversos equipamentos utilizados na automação industrial.

Destacou-se, em particular, a Especificação de Mensagem da Manufatura (MMS), por ser de fundamental importância para a interoperabilidade de equipamentos heterogêneos nos sistemas de produção. Salientamos como grande contribuição oferecida pelo MMS, a definição do Dispositivo de Manufatura Virtual (VMD), por ser um padrão a ser seguido nos equipamentos de produção. Também a adoção dos serviços MMS é fundamental para ter-se alto grau de flexibilidade computacional, devido as possibilidades de criação, destruição e modificação dinâmica dos vários objetos definidos no VMD.

Finalmente, foram apresentados os serviços MMS como interface entre o Sistema de Supervisão e o Sistema de Comando de uma Célula Flexivel de Usinagem, situando os conceitos apresentados em um exemplo prático.

CAPITULO 4

O PROTOCOLO MMS

4.1. Introdução

As interações das entidades da Camada de Aplicação são regidas por protocolos que definem as regras de comunicação e o formato das mensagens trocadas. Neste trabalho, estamos interessados nas interações efetivadas através do protocolo MMS.

O estudo do protocolo MMS e a análise das suas propriedades está baseado na obtenção do modelo do mesmo e na posterior validação deste modelo. Por razões de simplicidade de uso, de adequação ao problema de comunicação e de disponibilidade de ferramentas automáticas para análise, escolheu-se a Rede de Petri como ferramenta de descrição formal do protocolo MMS.

Para obtenção do modelo global do protocolo, adotou-se uma metodologia de decomposição modular em componentes representando o comportamento local das entidades de protocolo, interconectados através do modelo do serviço da camada inferior. A validação do protocolo através da prova da existência de um

conjunto de propriedades (limitação, ausência de interbloqueamento, etc.) e da verificação da correção do protocolo, permite demonstrar que o mesmo implementa efetivamente o serviço para o qual foi projetado.

Neste capítulo, será mostrado como o protocolo MMS pode ser validado, a partir da análise dos resultados obtidos sobre o seu modelo global, representado em Rede de Petri.

Devido ao fato dos serviços MMS serem orientados a associação, a análise completa do protocolo poderá ser feita por partes para cada conjunto de interações associadas às fases de estabelecimento do contexto, transferência de dados e abandono do contexto MMS.

4.2. Descrição do Protocolo MMS

4.2.1. Apresentação Geral do Protocolo MMS:

No protocolo MMS, as interações entre usuários não são simétricas, pois um lado do sistema de comunicação, chamado cliente, utiliza o outro lado, chamado servidor, para a realização de alguns serviços. O usuário MMS é considerado "servidor" quando pode ser modelado através de um Dispositivo de Manufatura Virtual (VMD), conforme visto no capítulo anterior.

As interações entre os usuários MMS são realizadas através dos serviços MMS, cujas regras e convenções adotadas no protocolo são fornecidas por uma parte da Entidade de Aplicação, chamada provedor MMS, que é composto por entidades de protocolo que trocam as Unidades de Dados de Protocolo MMS (PDU´s MMS), fazendo uso dos suporte oferecido pelos serviços das camadas inferiores.

4.2.2. As Unidades de Dados de Protocolos (PDU)

Para que as PDU´s possam ser compreendidas pelas entidades do protocolo, é necessário que tenham uma representação comum a nível de descrição dos seus dados.

Os dados apresentam-se sob dois aspectos:

- A sintaxe abstrata, que corresponde a descrição formal das estruturas genéricas de dados, conforme apresentado nas normas;

- A sintaxe de transferência, que indica a forma real de reagrupamento dos dados, durante a transmissão.

A Especificação de Mensagem da Manufatura utiliza a Notação de Sintaxe Abstrata Um (ASN.1), enquanto a sintaxe de transferência é determinada através da negociação pelo Serviço de Apresentação, sendo que deverão ser suportadas, além de ASN.1, outras sintaxes de transferências que podem ser negociadas.

A maior parte dos serviços MMS possuem parâmetros que são passados entre os usuários, por meio de trocas de primitivas de serviço com o provedor MMS, sem que seja feita qualquer alteração nesses parâmetros. O provedor MMS constrói as PDU´s baseadc nos parâmetros da primitiva de Pedido do serviço, recebida do usuário requisitor, e as envia para a entidade remota, através dos serviços oferecidos pela Camada de Apresentação. A entidade remota recebe esta PDU e passa os

parâmetros para o usuário respondedor, através da primitiva de Indicação do serviço. Quando o serviço é confirmado, a entidade remota aguarda a recepção de uma primitiva de Resposta de serviço do usuário respondedor, passando-a, através da camada de Apresentação, para a entidade que enviou a PDU de pedido. Os parâmetros desta PDU são então passados para o usuário requisitor através da primitiva de Confirmação de serviço. O cenário representativo destas duas situações foi apresentado na figura 4.

Caso o parâmetro da primitiva de Resposta tenha um resultado positivo, indicando que o serviço foi realizado, o provedor envia uma PDU de confirmação, caso contrário envia uma PDU de erro, indicando explicitamente o motivo do serviço não ser realizado.

4.2.3. Descrição do Protocolo MMS

Uma vez que os serviços MMS são orientados a Associação, o protocolo MMS é constituído por três fases distintas: Estabelecimento do Contexto MMS, Tranferência de Dados e Abandono do Contexto MMS.

Inicialmente, o contexto MMS é negociado através do serviço "Initiate", como pode ser visto pela figura 21, onde é mostrado o diagrama de estado para entrar e sair do Ambiente MMS. Dentro do Ambiente MMS os usuários podem fazer uso de qualquer serviço disponível, obedecendo as regras do protocolo MMS. Após sair deste Ambiente, através do serviço "conclude" ou "abort", não mais poderá ser usado qualquer serviço, até que o contexto MMS seja novamente negociado.

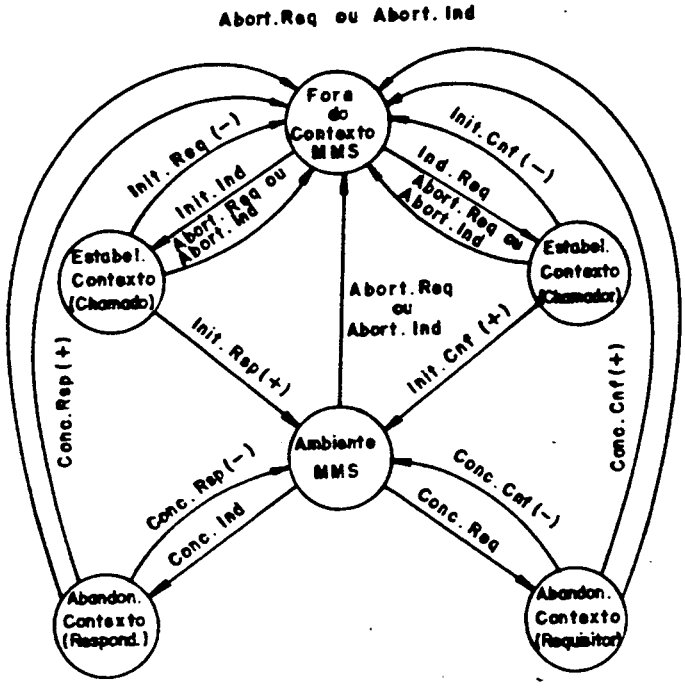


Fig. 21: Entrada e Saída do Ambiente MMS

A seguir, será apresentado o protocolo correspondente aos serviços de Gerenciamento de Contexto.

4.2.3.1. Estabelecimento do Contexto MMS

O serviço "Initiate" permite a um usuário MMS iniciar o diálogo com outro usuário MMS, alocando os recursos que serão suportados pela comunicação, como também negociando alguns parâmetros, tais como: número da versão do MMS a ser utilizada, tamanho máximo das PDU's, número máximo de serviços pendentes no usuário chamado e no usuário chamador, nível de aninhamento nas estruturas de dados, além da identificação dos serviços suportados.

O usuário chamador é o que inicia a comunicação através de um pedido de serviço "Initiate" ao provedor, que constrói a PDU correspondente e a envia para a entidade homóloga.

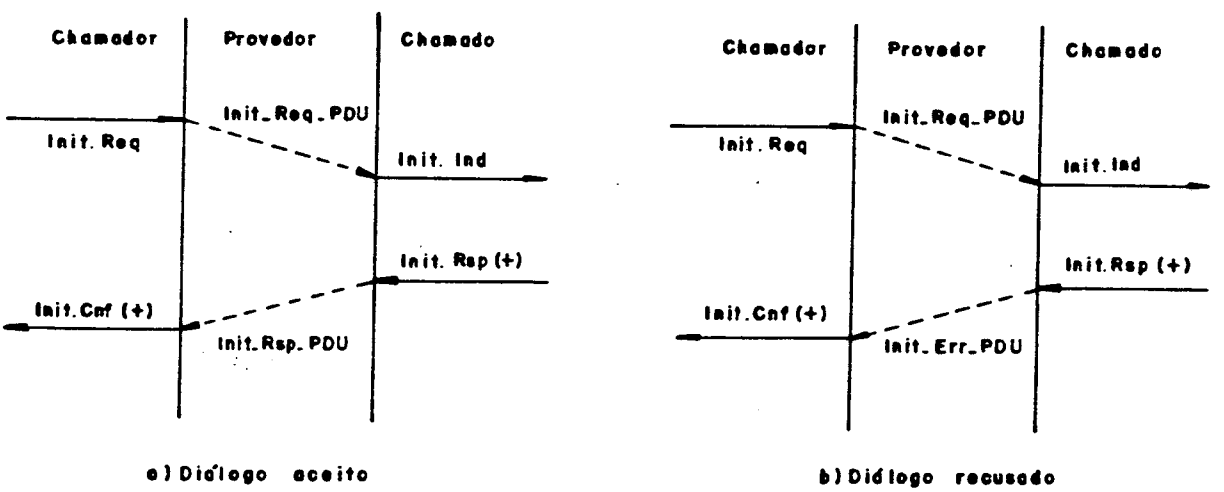


Fig. 22: Execução do serviço "Initiate"

Após receber a primitiva de Indicação do serviço "Initiate", do provedor MMS, o usuário chamado deverá emitir a primitiva de Resposta contendo um resultado positivo, se estiver desejando estabelecer o contexto MMS sob as condições indicadas, e com um resultado negativo, caso contrário. A figura 22 apresenta a troca de PDU do protocolo correspondente ao serviço Initiate.

A execução com sucesso do serviço "Initiate", resulta no estabelecimento do contexto MMS, onde qualquer outro serviço pode ser utilizado, exceto o serviço "Initiate". A partir de então, o usuário que desejar utilizar um serviço passa a ser chamado de requisitor e o usuário que atender a este pedido, de respondedor.

4.2.3.2. Transferência de Dados

Na fase de tranferência de dados pode-se utilizar quaisquer serviços MMS, sejam eles confirmados ou não.

Para cada pedido de serviço existe um Objeto Transação, o qual governa o processamento do mesmo; este objeto existe somente durante o tempo necessário para a execução do serviço. Associado com cada objeto há um identificador de Invocação ("Invoke_Id") que identifica univocamente cada instância de pedido de serviço.

Em qualquer instante podem existir múltiplas instâncias de pedidos de serviços confirmados pendentes, que ainda não receberam confirmação. Estes serviços podem ser cancelados pelo usuário requisitor através do pedido de serviço "Cancel".

Além do serviço "cancel", outro Serviço de Gerenciamento de Contexto que é utilizado na fase de Transferência de Dados é o serviço "Reject", que é utilizado pelo provedor do protocolo para notificar aos usuários, a ocorrência de erros de protocolo. Estes serviços serão apresentados a seguir.

Serviço "Cancel"

Este serviço é do tipo confirmado e não afeta o número máximo de serviços pendentes, devendo existir no máximo uma única invocação de pedido deste serviço para cada instância de serviço confirmado. O único serviço que não pode ser cancelado é, obviamente, o próprio serviço "Cancel" e os serviços não confirmados.

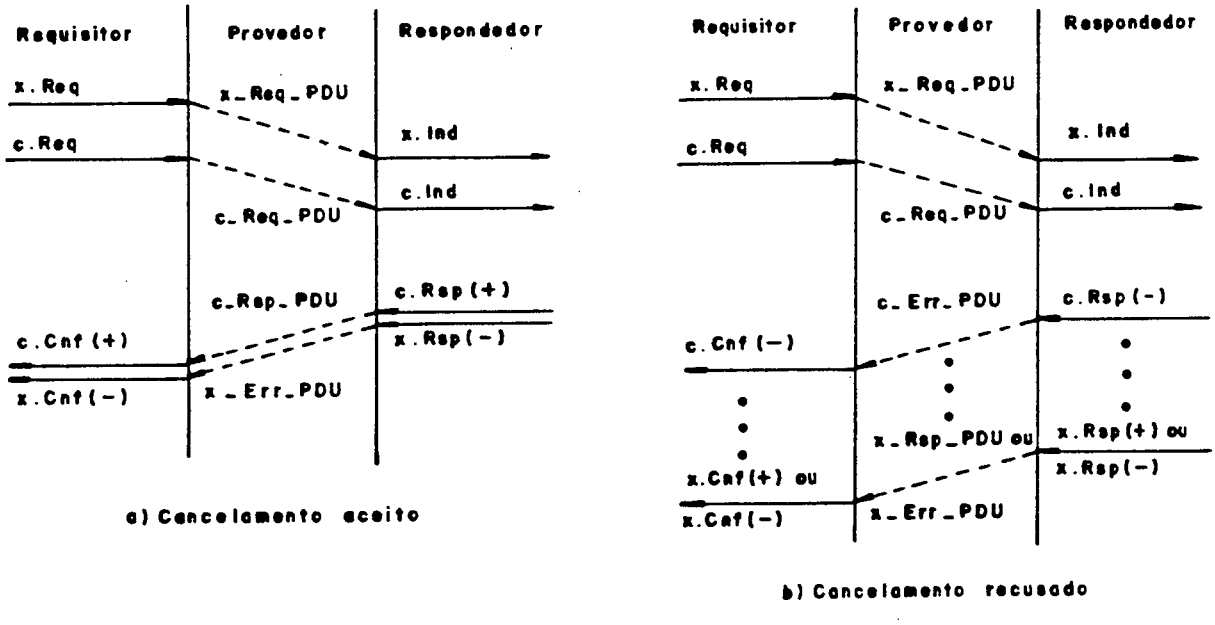


Fig. 23: Cancelamento de um serviço confirmado

O processamento deste serviço é idêntico à de qualquer serviço confirmado e cabe ao usuário respondedor aceitar ou não o cancelamento de um serviço determinado; porém se o cancelamento do serviço for aceito pelo usuário respondedor, este deverá enviar ao provedor uma primitiva de Resposta, confirmando o cancelamento do serviço, como também, uma primitiva de Resposta indicando que o serviço não foi realizado devido ao seu cancelamento. A figura 23 apresenta a troca de PDU no caso de um cancelamento de serviço confirmado ser aceito ou recusado.

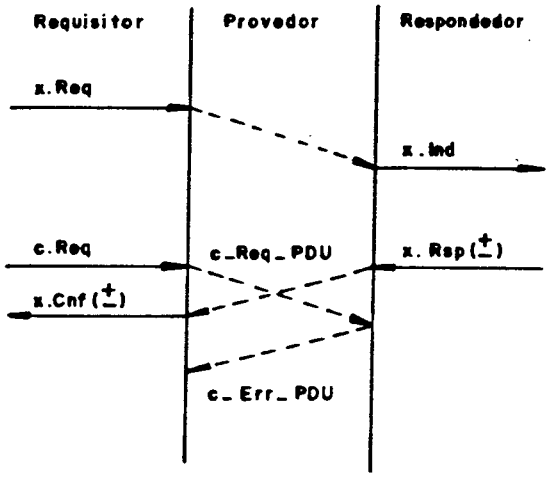


Fig. 24: Cancelamento recusado pelo provedor

Quando o provedor recebe uma PDU de pedido de cancelamento tentando cancelar um pedido de serviço desconhecido, este deverá enviar uma PDU de erro de cancelamento ao usuário que solicitou o serviço "Cancel", sem que o usuário respondedor seja notificado. Isto ocorre quando uma entidade envia a PDU de pedido de cancelamento (C_Req_PDU) simultaneamente com o envio, pela outra entidade, das PDUs de resposta do serviço confirmado (x.Rsp.PDU ou x.Err_PDU), conforme apresentado na figura 24.

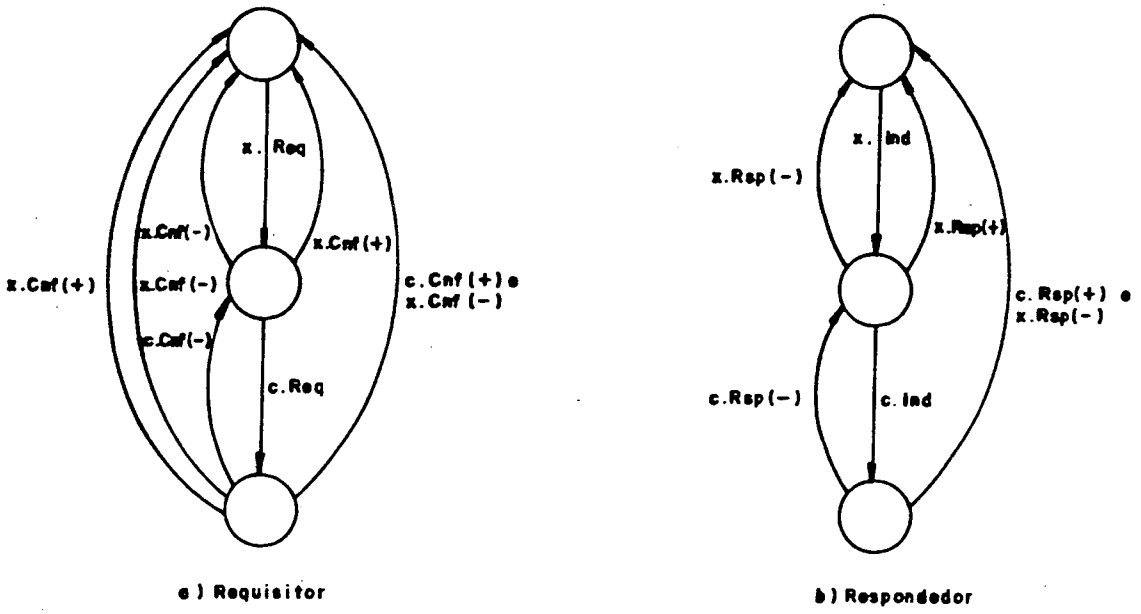


Fig. 25: Pedido e atendimento de serviço confirmado

Podemos observar na figura 25 onde é mostrado o diagrama de estado de pedido de serviço confirmado, que após a confirmação (x.cnf(+) ou x.cnf (-)) do serviço aguardando cancelamento no requisitor, esta instância de pedido de serviço é considerada encerrada, mesmo que esteja aguardando cancelamento. Pelo fato do serviço "Cancel" ser um serviço confirmado, é de se esperar que, após a recepção de uma primitiva de Pedido deste serviço pelo provedor, este deva emitir uma primitiva de Confirmação do

serviço "Cancel" ao usuário requisitor, após ter recebido a PDU de resposta, mesmo que o outro usuário já tenha concluído a execução do serviço.

Serviço "Reject"

A ocorrência de erros de protocolos é notificada ao usuário, através do serviço "Reject", para que o mesmo possa decidir sobre a continuação do diálogo, que poderá ser encerrado através do serviço "Abort", descrito na fase de abandono de contexto MMS. O serviço "Reject" é somente iniciado pelo provedor que o envia ao usuário através da primitiva de Indicação de serviço e para a entidade remota através de uma PDU de rejeição ("reject_PDU"), como dado de usuário no serviço da Camada de Apresentação, conforme apresentado na figura 26.

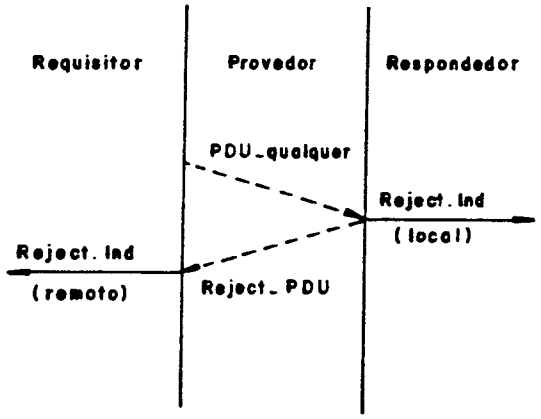


Fig. 26: Erros do protocolo

O erro de protocolo pode ter origem na estrutura da PDU (desconhecida, incorreta ou inválida) ou na semântica associada a mesma como identificador de invocação duplicado ou não reconhecido, serviço não reconhecido, comprimento excessivo da PDU, valor fora da faixa permitida, nível de recursão excessivo, número máximo de invocações de pedido de serviço ultrapassado, entre outros

4.2.3.3. Abandono do Contexto MMS

O usuário pode encerrar o diálogo estabelecido de forma negociada através do envio de uma primitiva de pedido de serviço "Conclude" ao provedor MMS, ou de forma abrupta, não negociada, através do serviço "Abort", liberando, desta forma, todos os recursos específicos da Associação de Aplicação alocados para esta comunicação.

Serviço "Conclude"

Após ter sido iniciado o procedimento de serviço "Conclude", o usuário requisitor não poderá emitir qualquer outra primitiva de Pedido de serviço até ter recebido a confirmação do serviço "conclude". Se a confirmação for positiva, o contexto MMS é considerado encerrado, caso contrário não será afetado.

Após ter recebido a primitiva de Indicação do Serviço "Conclude" do provedor MMS, o usuário respondedor não poderá emitir qualquer outra primitiva de serviço até aceitar ou não a conclusão, por meio de uma primitiva de Resposta do serviço "Conclude". Se esta primitiva possuir um resultado positivo, o contexto MMS é considerado terminado e não mais poderá ser emitido primitivas neste contexto. Neste caso, o usuário respondedor não deverá estar esperando qualquer resposta do usuário homólogo para um serviço confirmado. Por outro lado, se a primitiva possuir um resultado negativo, o contexto MMS não será afetado e a emissão de primitivas continua normalmente.

Serviço "Abort"

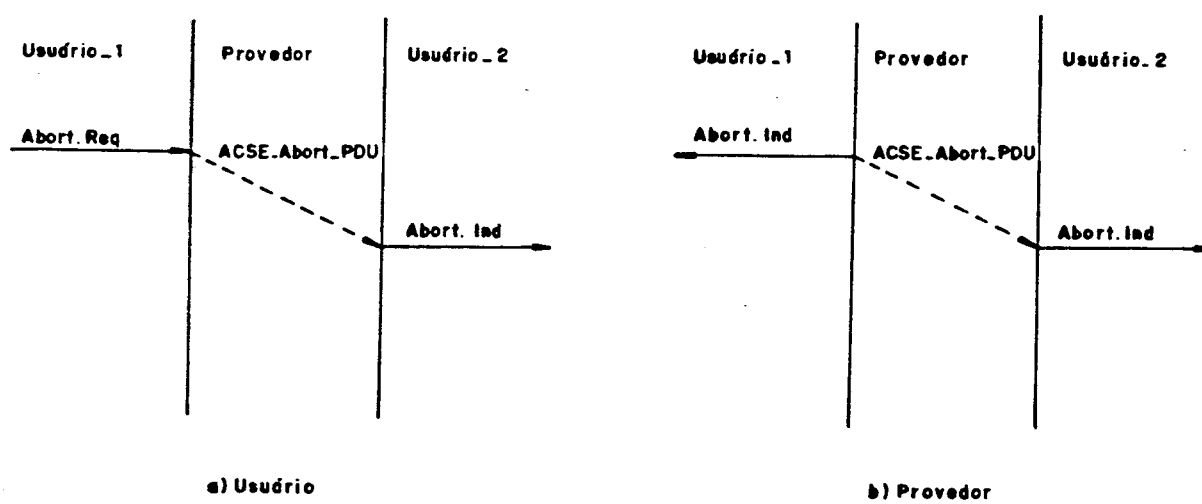


Fig. 27: Serviço "Abort"

O serviço "Abort" é diretamente mapeado no serviço "A-Abort" do ACSE, uma vez que MMS não define uma PDU específica para o mesmo. Este serviço pode também ser emitido pelo provedor MMS, especialmente quando ocorrem erros não recuperáveis na comunicação, neste caso ambos os usuários são notificados através de uma primitiva de Indicação, conforme pode ser visto na figura 27. O serviço "abort" é o único serviço que está sempre disponível para ser usado, a partir do pedido do serviço "Initiate".

Se o serviço "Initiate" ainda não tiver sido confirmado, e houver a recepção de uma PDU inválida ou outro erro de protocolo, o provedor deverá emitir uma primitiva de Indicação do serviço "abort" ao usuário MMS e emitir uma primitiva de Pedido "A-Abort. Req" ao ACSE, se uma Associação de Aplicação existe.

4.3. Modelo Global do Protocolo MMS

O modelo global de um protocolo é obtido pela interconexão, através do serviço da camada inferior, de duas entidades de protocolo. Foi adotado um modelo para representar cada uma das fases de funcionamento do Protocolo MMS, descritas no item anterior. A seguir são mostrados os modelos das entidades de protocolos correspondentes as Fases de Estabelecimento do Contexto MMS, além do modelo da camada inferior para interconectar as duas entidades de protocolo.

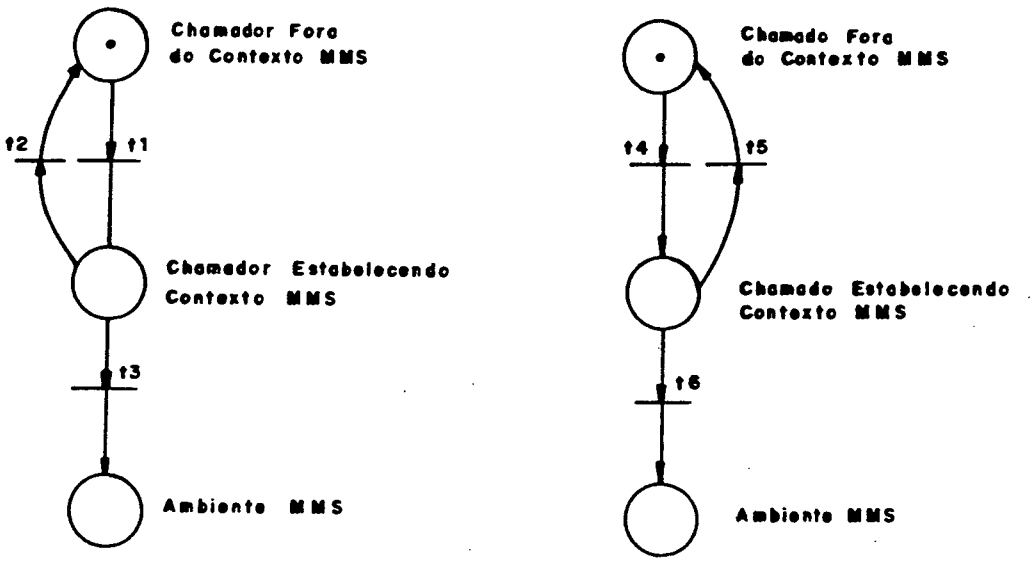
4.3.1. Modelo das Entidades de Protocolos

O comportamento local de cada entidade de protocolo é especificado por uma Rede de Petri predicado/ação. A comunicação existente entre duas entidades homólogas são explicitamente indicadas no modelo por uma transição associada com o envio de uma mensagem (!M), juntamente com outra transição, na outra entidade, associada com a recepção da mesma mensagem (?M). Além disso, as transições são etiquetadas da forma Predicado/Ação, onde "Predicado" deve ser satisfeito para que a transição possa disparar e a "Ação" possa ser executada.

Durante a fase de Estabelecimento de Contexto MMS a entidade de protocolo é descrita através de um dos dois modelos seguintes apresentados na figura 28:

"chamador": que inicia a comunicação com o outro usuário MMS, de forma a estabelecer o contexto MMS.

"chamado" : que aceita ou não o estabelecimento do contexto MMS.



t1 - ?Init.Req / !Init_Req_PDU

t2 - ?Init_Err_PDU / !Init.cfn(-)

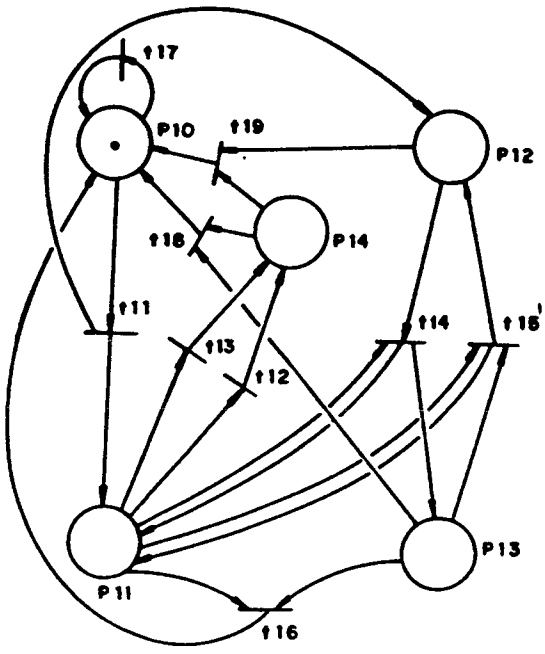
t3 - ?Init_Rsp_PDU / !Init.cfn(+)

t4 - ?Init_Req_PDU / !Init.Ind

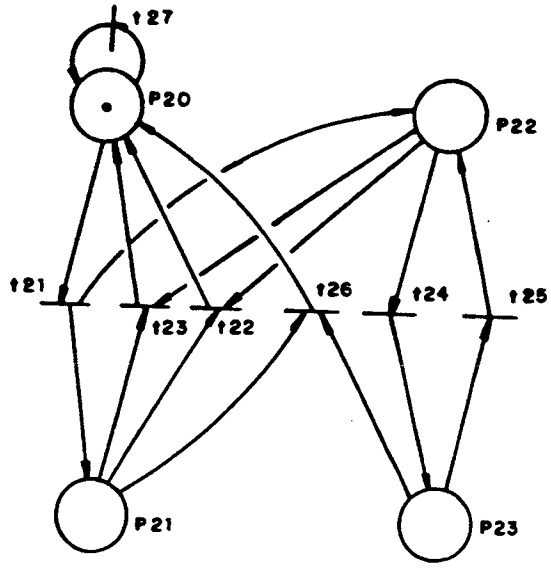
t5 - ?Init.Rsp(+) / !Init_Err_PDU

t6 - ?Init.Rsp(-) / !Init_Rsp_PDU

Fig. 28: modelo da entidade chamadora e chamada



a) Requisitor



b) Respondedor

P10= Requisitor Inativo
P11= Esperando Reconhecimento
P12= Pronto para Cancelar
P13= Aguardando Cancelamento
P14= Desfazendo Cancelamento

P20= Respondedor Inativo
P21= Esperando Resposta do Usuário
P22= Disponível para Cancelar
P23= Aguardando Resposta ao Cancelamento

T11: ?x.Req / !x_Req_PDU
T12: ?x_Rsp_PDU / !x.Cnf(+)
T13: ?x_Err_PDU / !x.Cnf(-)
T14: ?c.Req / !C_Req_PDU
T15: ?c_Err_PDU / !c.Cnf(-)
T16: ?c_Rsp_PDU e ?x_Err_PDU
!c.Cnf(+) e !x.Cnf(-)
T17: ?c_Err_PDU /

T21: ?x_Req_PDU / !x.Ind
T22: ?x.Rsp (+) / !x_Rsp_PDU
T23: ?x.Rsp (-) / !x_Err_PDU
T24: ?c_Req_PDU / !c.Ind
T25: ?c.Rsp(-) / !c_Err_PDU
T26: ?c.Rsp (+) e ?x.Rsp (-)
!c_Rsp_PDU e !x_Err_PDU
T27: ?c_Req_PDU / !c_Err_PDU

Fig. 29: Entidades requisitora e respondedora

Para modelar a fase de transferência de dados, considera-se que uma entidade representa o requisitor do serviço e a outra entidade, o respondedor do serviço, conforme apresentado na figura 29.

Este modelo representa uma única instância de serviço confirmado pendente. No caso geral devemos ter tantos modelos desta entidade quanto são o número de pedidos de serviços pendentes negociados através do serviço "Initiate". Por simplificação, considera-se para efeito de análise, um único serviço pendente negociado; modelos mais elaborados tais como Redes de Petri Predicado/Transição poderão permitir uma representação compacta dos vários serviços pendentes.

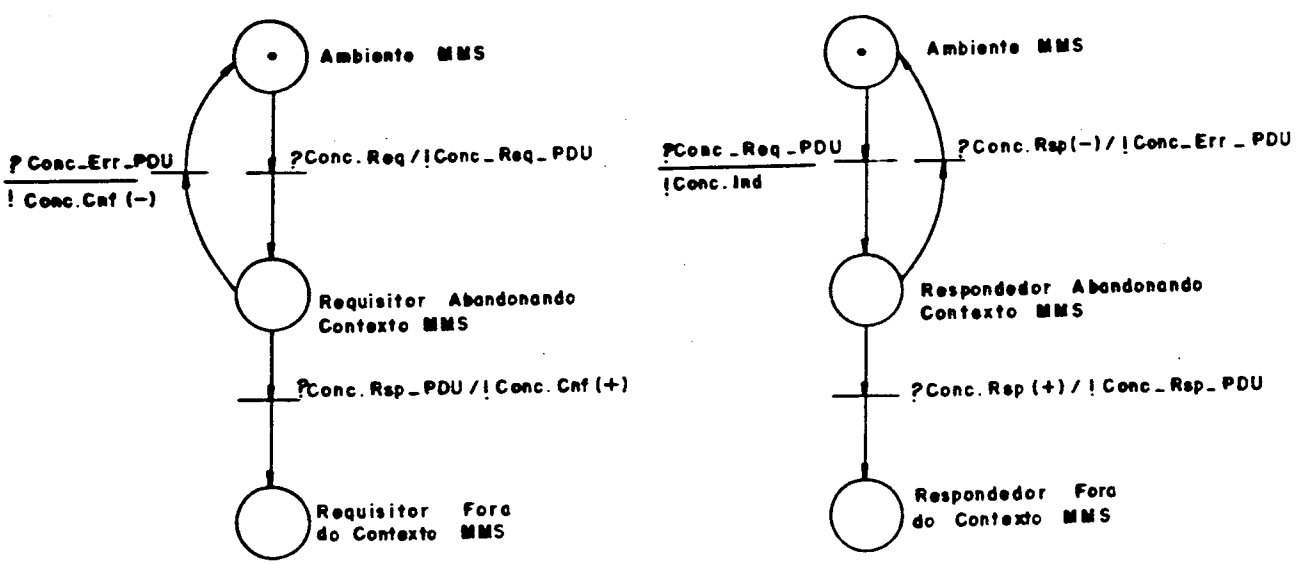


Fig. 30: Saída do Ambiente MMS

A fase de saída negociadada do ambiente MMS é modelada pelas entidades requisitora e respondedora, as quais respectivamente pedem o encerramento das transações entre as entidades e aceitam ou não este encerramento (figura 30).

4.3.2. Interconexão das Entidades de Protocolo:

O modelo global do protocolo é obtido interconectando-se cada modelo das entidades locais, através do serviço fornecido pela camada inferior. As interligações poderão ser feitas por "acoplamento direto" ou através da representação de um "meio virtual".

No acoplamento direto, as interações são modeladas pela fusão das transições associadas ao envio e recepção de uma mesma mensagem (!M com ?M) conforme figura 31 Neste tipo de acoplamento considera-se o meio de comunicação perfeito entre as entidades comunicantes. Porém, tem como inconveniente a necessidade de inclusão de transições de emissão adicionais, quando a mensagem pode ser recebida em mais de um estado da entidade recebedora e, de forma idêntica, a inclusão de transições de recepção adicionais quando uma mensagem pode ter diversas origens, na entidade emissora.

Por esta razão e apesar da sua simplicidade, a utilização do acoplamento direto não é a mais conveniente para interconectar as duas entidades, pois as transições adicionais não representam a comunicação real. Entretanto, é perfeitamente aplicável para interconectar as diversas atividades dentro de um mesmo sistema.

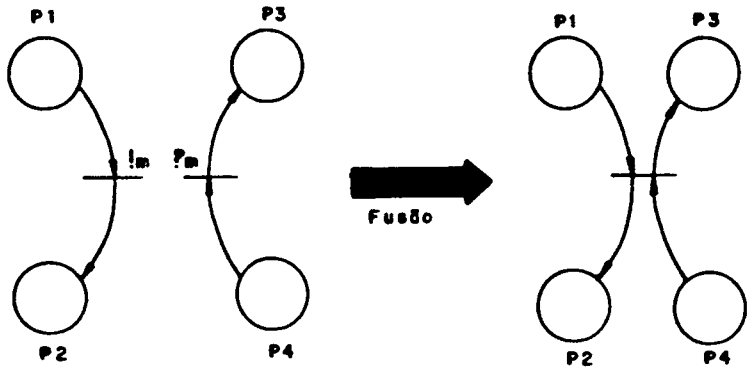


Fig. 31: Fusão de transição (meio ideal)

A interconexão através de um "meio virtual" é uma abordagem que permitirá obter um modelo global mais próximo da realidade, permitindo representar comportamentos tais como perda de mensagem, duplicação da mensagem, inclusão de erros na mensagem, ordenamento FIFO, tamanho limitado ou ilimitado do meio, etc.. O meio virtual será explicitamente modelado através de Redes de Petri predicado/ação para representar as mensagens em trânsito dentro do meio de comunicação.

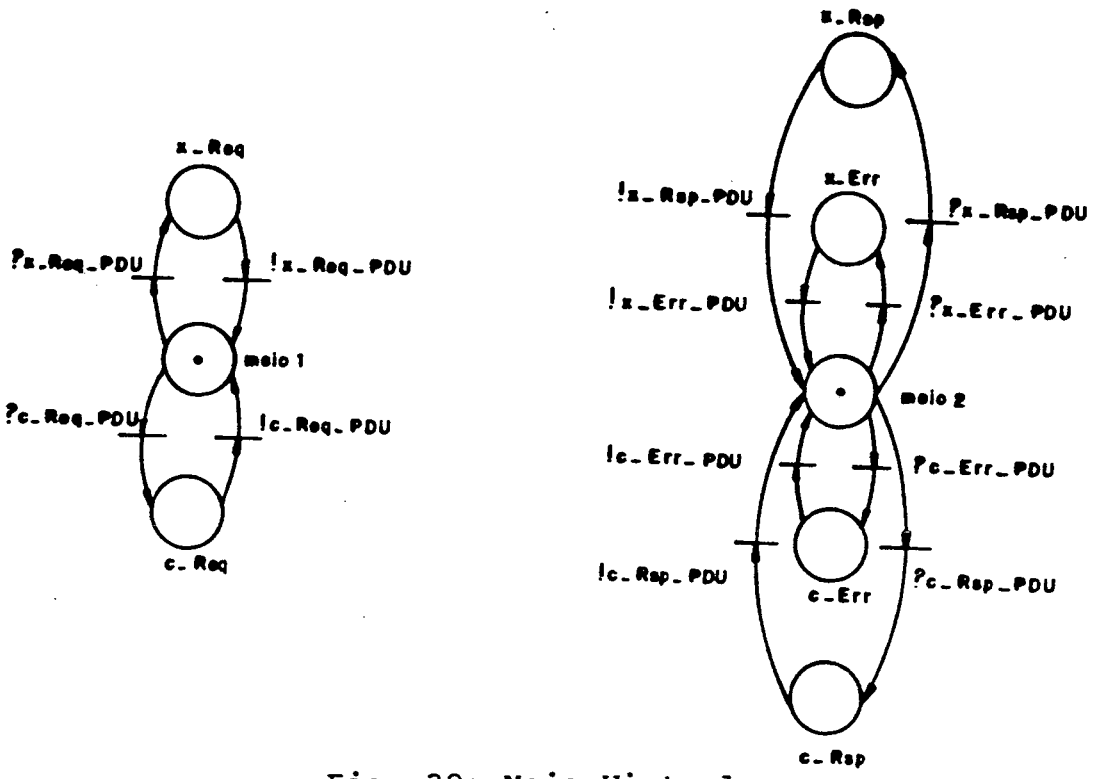


Fig. 32: Meio Virtual

Entretanto, como o serviço Full-Duplex, ou seja, transmissão simultânea em dois canais, é sempre usado, o MMS necessita da Unidade Funcional Duplex da camada de Sessão, por isso, o protocolo MMS não considera erros de transmissão, mensagens fora de ordem, perdas de mensagens ou mensagens duplicadas, uma vez que estes problemas serão resolvidos pelas camadas inferiores.

Também não há nenhum mecanismo de controle de fluxo explícito. Para limitar a emissão de dados, a entidade respondedora utiliza o controle de fluxo implicito ("Flow Control by Back Pressure") sobre a Associação de Aplicação. O controle de fluxo implícito é unicamente a repercussão, para os usuários do serviço, dos efeitos do "controle de fluxo" definido a nível de protocolo; isto corresponde ao número de pedidos de serviços confirmados pendentes, negociado no estabelecimento do "contexto MMS", através do serviço "Initiate".

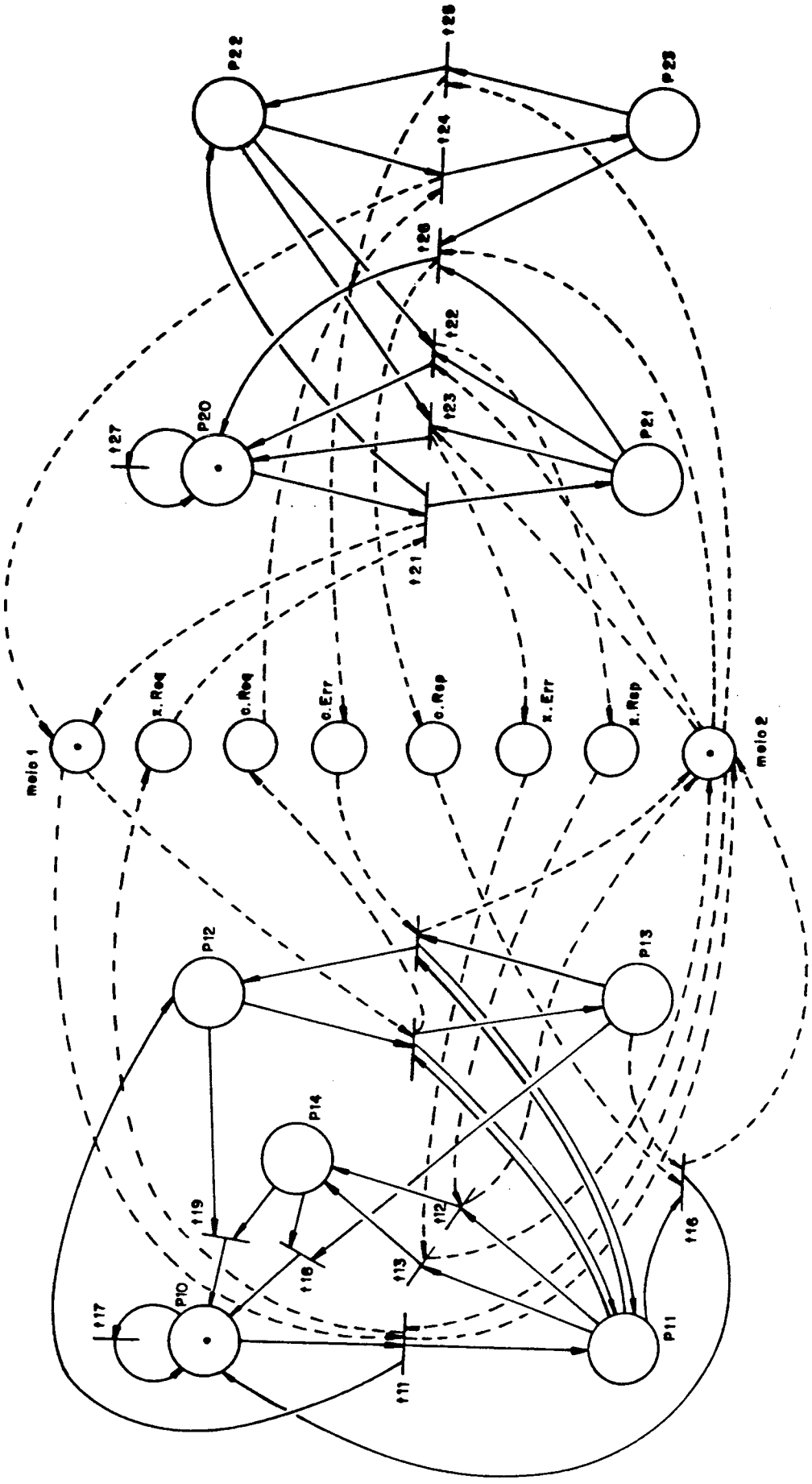


Fig. 33: Modelo global

Em conseqüência adotou-se no modelo um lugar dedicado a cada tipo de mensagem transmitida (x_Req, x_Rsp, x_Err, c_Req, c_Rsp, c_Err); as fichas nestes lugares indicam a presença da mensagem especificada dentro do meio virtual. para limitar o número de mensagens no meio e o ordenamento das mensagens permitimos apenas uma única mensagem em trânsito para cada sentido da comunicação, conforme figura 32, onde "meio1" representa o meio de comunicação livre na direção Requisitor para Respondedor, enquanto "meio2" representa o meio livre na direção contrária.

Este modelo será acoplado diretamente às outras entidades através da fusão de transições associadas ao envio de mensagem (!M) e a recepção de mensagens (?M), fornecendo o modelo global, esquematizado para a fase de transferência de mensagens na figura 33. As transições do modelo do meio que devem ser acopladas a mais de uma transição do modelo das entidades serão duplicadas, antes de haver a fusão dos modelos.

4.4. Análise do Protocolo MMS

A análise do protocolo MMS, modelado por Rede de Petri, envolve a utilização dos resultados referentes as propriedades gerais e específicas da rede de Petri, obtidos através de ferramentas de análise. Foi utilizado neste trabalho o Analisador de Rede de Petri ARP, desenvolvido no LCMI-EEL-UFSC, o qual possibilita obter as propriedades da rede através da enumeração de marcações acessíveis e da determinação dos invariantes, além da simulação passo a passo e da verificação através do método de projeção.

Foi estabelecido anteriormente que o modelo proposto podia ser dividido em três fases (estabelecimento de contexto, transferência de dados, saída do contexto) que podiam, numa primeira abordagem, serem analisadas separadamente. Cada uma das fases foi analisada através do método descrito no capítulo 2, destacando a prova das propriedades gerais que indicam para um protocolo qualquer o seu perfeito funcionamento e das propriedades específicas associadas a semântica do protocolo MMS.

Entretanto, para efeito de apresentação do método somente a fase de transferência de dados será discutida, por ser a de maior grau de complexidade; para as fases de estabelecimento e de saída de contexto, verificou-se facilmente a correção do protocolo em relação ao serviço especificado. Além disso, para a fase de transferência de dados, considera-se apenas o aspecto de funcionamento normal do protocolo, sem possibilidade de ocorrências de exceções, como perda da comunicação, erros de protocolo ("Reject") e encerramento abrupto do diálogo ("Abort"); restringe-se também, ao mínimo, o número de pedidos de serviços pendentes, já que cada instância de pedido é univocamente identificada por um identificador de invocação ("InvokeId").

4.4.1. Propriedades Gerais

A partir da análise do modelo global do protocolo MMS (fase de transferência de dados), apresentado no item anterior verifica-se que a rede é binária (ou seja, limitada a uma ficha no máximo por lugar), viva e reiniciável, o que permite concluir que o protocolo possui todas as características desejadas de limitação de recursos, de vivacidade em relação a todos os eventos modelados e de ciclicidade.

4.4.2. Propriedades Específicas

Estas propriedades são dependentes da semântica associada aos lugares e transições da rede que modela o protocolo considerado, permitindo interpretar os invariantes de lugar e de transição, obtidos a partir do modelo global.

4.4.2.1. Correção Parcial do Protocolo

A correção parcial do protocolo pode ser provada verificando-se, através da interpretação dada aos invariantes de lugar, as asserções que descrevem o comportamento do protocolo ou comparando-se o modelo do serviço fornecido pelo protocolo, reduzido através do método de projeção, com o modelo do serviço desejado. Estas duas metodologias foram apresentadas no capítulo 2.

Neste trabalho, escolhemos o método de projeção para a verificação da correção parcial do protocolo, devido a forma sistemática com que se pode utilizá-lo. É muito mais simples compararmos o modelo do serviço fornecido pelo protocolo com o modelo do serviço desejado, do que definirmos as asserções relativas ao comportamento do mesmo. Entretanto, o uso dos invariantes de lugar podem também ser utilizado para obter informações à respeito das máquinas de estado que compõe o protocolo, facilitando o seu entendimento interno e a localização de eventuais erros.

Comparação de Modelos: Método de Projeção

Para usarmos este método temos que ter um modelo explícito do serviço desejado para ser comparado com o modelo reduzido do serviço fornecido pelo protocolo, o qual é obtido, considerando-se como eventos visíveis a recepção e envio de primitivas de serviço. Desta forma, temos que considerar os eventos visíveis associados ao usuário requisitor e ao usuário respondedor (figura 34).

Os eventos visíveis, do ponto de vista do usuário requisitor, são o envio das primitivas de Pedido do serviço confirmado (x.Req) e do serviço "Cancel" (c.Req), a recepção das primitivas de Confirmação do serviço confirmado (x.Cnf(+) e x.Cnf(-)) e do serviço "cancel" (c.Cnf(+) e c.Cnf(-)). Esses eventos estão associados as transições t11, t12, t13, t14, t15 e t16 da Rede de Petri.

Do outro lado, do ponto de vista do usuário respondedor, os eventos visíveis são a recepção das primitivas de Indicação do serviço confirmado (x.Ind) e do serviço "cancel" (c.Ind), o envio das primitivas de Resposta do serviço confirmado (x.Rsp(+) e x.Rsp(-)) e do serviço "Cancel" (c.Rsp(+) e c.Rsp(-)). Esses eventos estão associados as transições t21, t22, t23, t24, t25 e t26 da Rede de Petri.

Os outros eventos são considerados eventos internos e não deverão aparecer na visão abstrata, obtida a partir da redução do grafo de marcações da rede de Petri global.

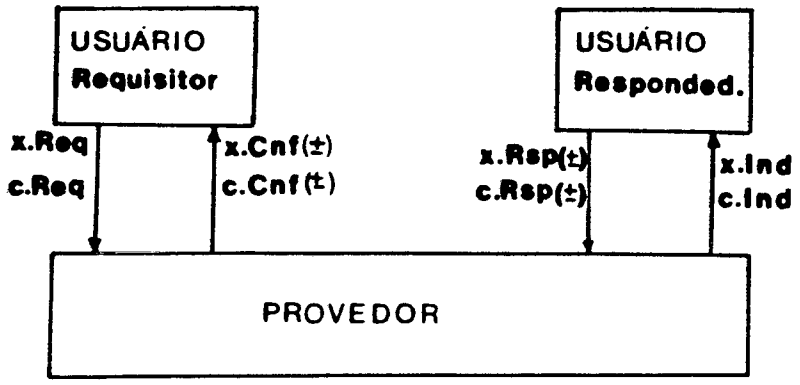


Fig. 34: Visibilidade do serviço

A figura 35 mostra o grafo de acessibilidade obtido da rede original, destacando os eventos visíveis pelo usuário requisitor e pelo usuário respondedor.

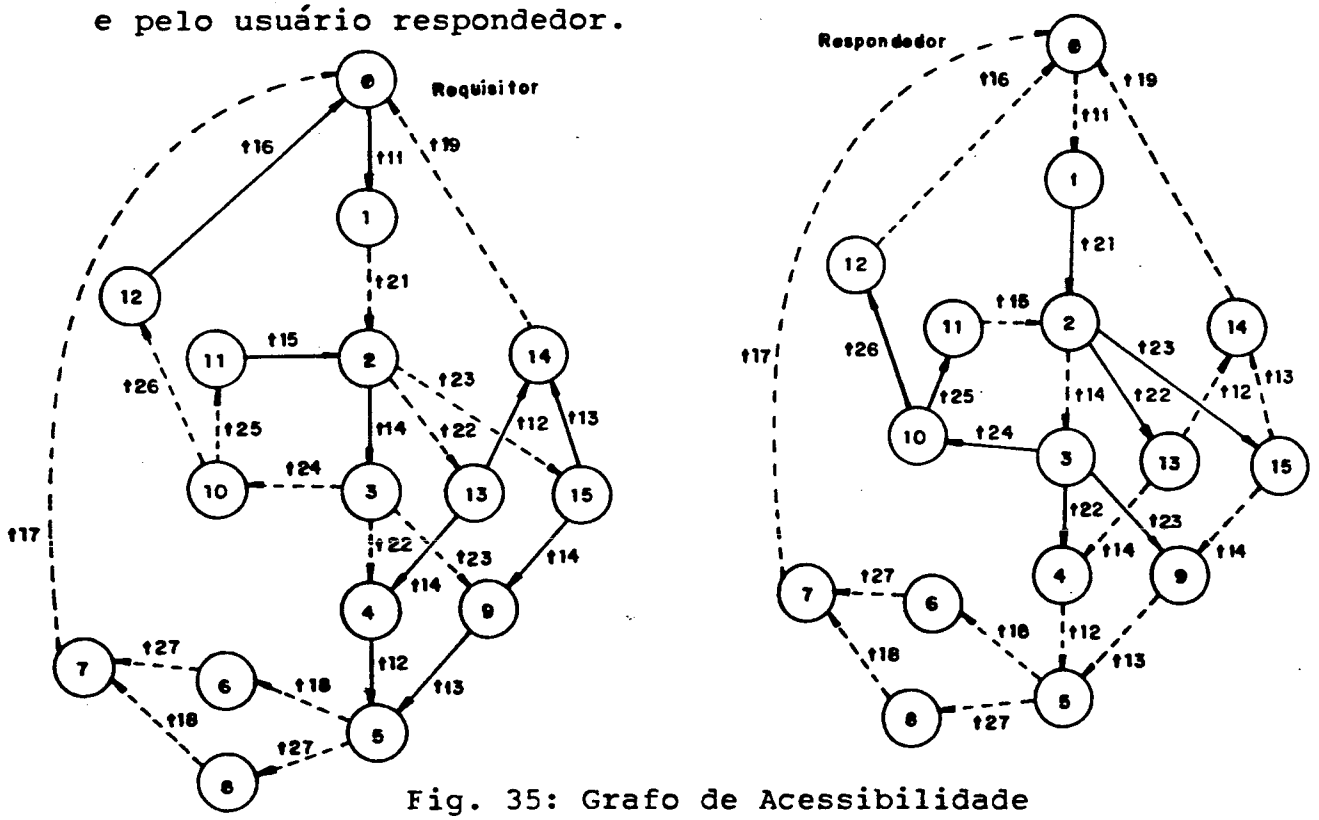


Fig. 35: Grafo de Acessibilidade

O uso do método de Projeção permite reduzir o modelo, através da eliminação das transições invisíveis e dos estados equivalentes, obtendo-se um autômato reduzido com poucos estados e que corresponde ao modelo do serviço efetivamente fornecido pelo protocolo.

A figura 36 mostra o resultado da redução dos grafos relacionados com o usuário requisitor e respondedor. Este modelo reduzido, obtido pelo método de projeção, deve ser comparado com o modelo do serviço desejado, diretamente a partir do autômato ou a partir da linguagem gerada por este.

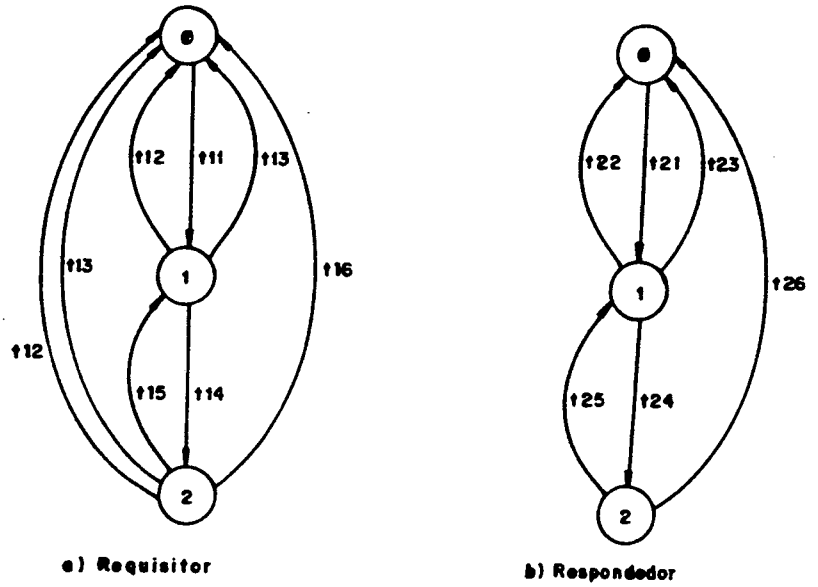


Fig. 36: Grafos reduzidos

Neste caso específico, foram comparados diretamente os autômatos reduzidos (figura 36) com a especificação do serviço fornecido pela norma ISO DP 9506, conforme apresentado na figura 25, concluindo-se que os mesmos possuem uma equivalência direta. Deste estudo, pode-se concluir que o protocolo está parcialmente correto e realiza efetivamente o serviço para o qual foi projetado.

4.4.2.2. Evolução do Protocolo

Para provar a correção global do protocolo, é necessário ainda mostrar que o protocolo evolue efetivamente.

Os invariantes de transição definem as seqüências cíclicas da rede, e podem ser utilizados para verificar a evolução do protocolo, determinando as possíveis seqüências infinitas não produtivas, que poderão resultar em bloqueamento no tempo ("Livelock").

Considerando-se o caso do serviço confirmado, foram analisados 6 invariantes de transição, destacados a seguir.

O primeiro invariante linear de transição (t11, t21, t22, t12, t19) corresponde a uma seqüência de pedidos de serviços confirmados que foram bem atendidos (x.req, x.Ind, x.Rsp(+), x.Cnf(+)) enquanto o segundo invariante (t11, t21, t23, t13, t19) corresponde a outra possibilidade, ou seja, uma seqüência possível de pedidos de serviço confirmado que não puderam ser atendidas (x.req, x.Ind, x.Rsp(-), x.Cnf(-)).

O terceiro invariante (t11, t21, t14, t24, t26, t16) diz respeito às seqüências que resultam no cancelamento bem sucedido (x.req, x.Ind, c.Req, c.Ind, c.Rsp(+) e x.Rsp(-), c.Cnf(+) e x.Cnf(-)) do serviço que está pendente; por outro lado, o quarto invariante (t14, t24, t15, t25) corresponde a uma seqüência de tentativas fracassadas em tentar cancelar um serviço (c.req, c.Ind, c.Rsp(-), c.Cnf(-)).

O quinto invariante (t11, t21, t14, t22, t27, t12, t18, t17) está associado a tentativa mal sucedida de cancelar um serviço que já tenha sido confirmado (x.req, x.Ind, c.Req, x.Rsp(+), x.Cnf(+)), devido ao tempo de transferência da PDU no meio virtual.

O último invariante linear de transição (t11, t21, t14, t23, t27, t13, t18, t17) está associado ao ciclo que trata de cancelar um serviço que já foi bem atendido (x.req, x.Ind, c.Req, x.Rsp(-), x.Cnf(-)).

Através da análise dos ciclos repetitivos descritos acima, podemos observar que não ocorrerá bloqueio no tempo ("Livelock") pois todos os ciclos são "produtivos", ou seja, possuem transições visíveis ao usuário. O exame dos invariantes mostram que todos os que possuem o pedido do serviço confirmado (x_Req), possuem também uma indicação (x.Ind), uma resposta (x_Rsp) e uma confirmação (x_Cnf) desse serviço. O único invariante que não possui estes dois tipos de pares de primitivas de serviço é o responsável pelos ciclos de cancelamento mal sucedidos, porém, o mesmo apresenta os pares equivalentes de primitivas para o serviço "Cancel" (c.req, c.Ind, c.Rsp(-), c.Cnf(-)).

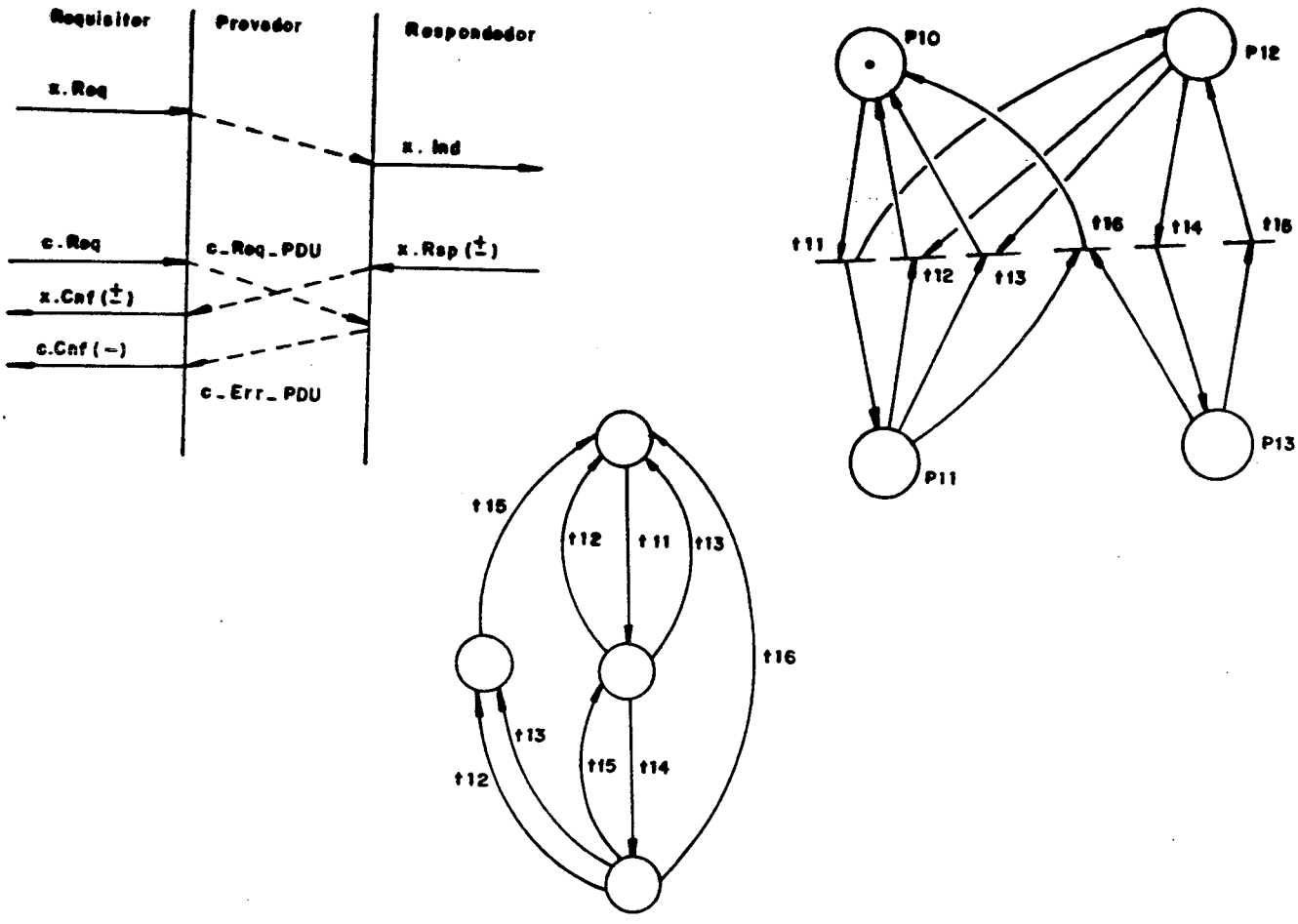


Fig. 37: Espera de PDU de erro

Além da <u>verificação da evolução</u> do protocolo, o estudo dos invariantes de transição permitiu-nos analisar alguns aspectos do comportamento do serviço "Cancel". Observa-se que, nos dois últimos invariantes de transição existe um pedido deste serviço (c.Req) sem a necessária confirmação (c.Cnf). Este caso ocorre quando a PDU de pedido de serviço "Cancel" alcança a entidade respondedora depois da mesma já ter enviado a PDU de resposta (ou de erro) do serviço a ser cancelado, conforme visto anteriormente neste capítulo. Esta situação foi apresentada na

figura 24 e nos leva a discutir o modelo do protocolo adotado pela norma ISO DIS 9506, a partir da consideração que para todo pedido de serviço confirmado é necessário esperar a confirmação.

Uma possível solução consiste em confirmar para o usuário a chegada da PDU de erro de serviço "Cancel", representada na figura 37, porém isto implicará em modificar o diagrama de estado que especifica o serviço confirmado, no nível do requisitor, conforme apresentado nesta mesma figura.

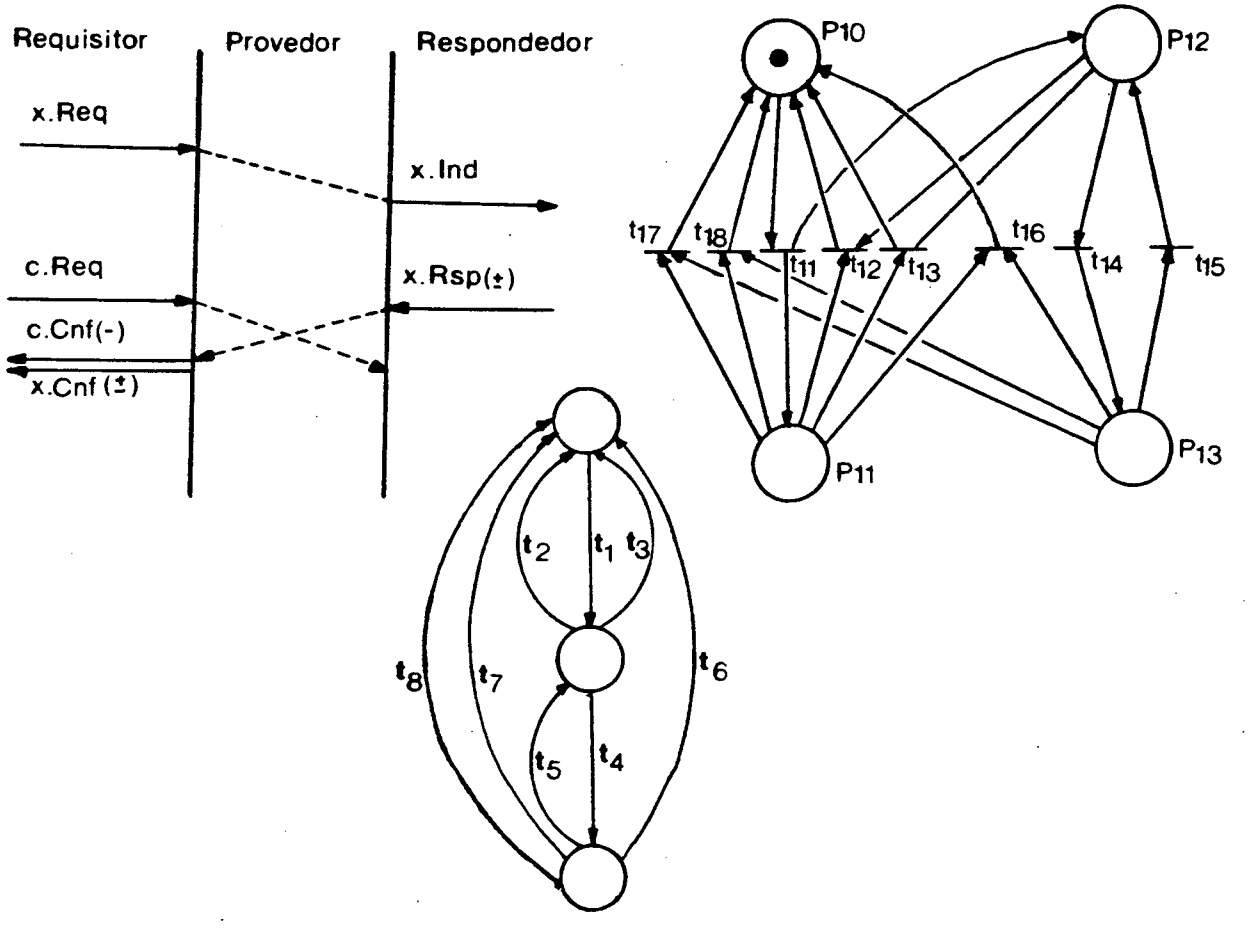


Fig. 38: Cancelamento pelo provedor

Esta solução tem o inconveniente de fazer com que o usuário fique esperando uma confirmação de cancelamento de um serviço que já foi atendido. Uma solução melhor consiste na confirmação do cancelamento pelo provedor (lado requisitor) juntamente com a confirmação do serviço, uma vez que estes eventos certamente irão ocorrer. Porém, nesta situação, não é mais necessário que a entidade respondedora envie a PDU de erro ao receber a PDU de cancelamento, basta apenas recebe-la e ignorá-la. A figura 38 apresenta o cenário, diagrama de estado e o novo modelo da entidade requisitora, de acordo com esta solução.

4.5 Conclusão

Neste capítulo, descreveu-se o protocolo MMS desconsiderando-se os aspectos ligados as estruturas de dados, que são fundamentais para a realização de uma aplicação. Constatou-se, a partir do estudo do protocolo MMS, a simplicidade do mesmo em comparação com os protocolos das camadas baixas, os quais necessitam, algumas vezes, de algorítmos complexos para a sua execução.

A seguir, foi obtido o modelo da parte do protocolo correspondente ao processamento dos serviços confirmados, fazendo uso da Rede de Petri como técnica de descrição formal. Os resultados obtidos pela análise do modelo permitiram validar facilmente o protocolo modelado. Verificou-se, em particular, a correção do protocolo e a adequação de sua implementação ao serviço desejado. A obtenção do modelo e a análise feita facilitaram o entendimento do comportamento da parte estudada do protocolo; a sistemática utilizada pode ser generalizada ao protocolo completo.

O exemplo apresentado visou mostrar as potencialidades da Rede de Petri na modelização e análise de protocolos. Neste estudo, entretanto, faltou abordar alguns aspectos importantes relacionados ao protocolo MMS, como a ocorrência de anormalidades (erros de protocolos, encerramento abrupto de

comunicação, etc) e instâncias múltiplas de serviços pendentes. Esta limitação foi devida principalmente a representação adotada e as ferramentas de análise disponíveis. Seria necessário em futuros trabalhos adotar uma outra representação com maior poder de representação, tal como Rede de Petri de Alto Nível (Predicado/Transição, Numérica) ou linguagem de especificação (Estelle), e ainda dispor de computadores com maiores capacidades para poder analisar o protocolo completo.

CAPÍTULO 5

CONCLUSÃO

Junto com as vantagens trazidas pela introdução dos Sistemas Informáticos Distribuídos nas aplicações industriais em termos de desempenho, disponibilidade, confiabilidade e flexibilidade, é necessário destacar as dificuldades ligadas aos aspectos de comunicação neste tipo de sistema. A complexidade deste tipo de problema requer, para ser tratada com eficiência e segurança, abordagens baseadas em conceitos de programação estruturada, tais como abstração de dados e refinamentos sucessivos, juntos com definições relacionadas com a arquitetura de sistemas.

A definição de uma estruturação hierarquizada em camadas para os Sistemas Distribuídos incorporou estes aspectos, proporcionando facilidades no projeto, na implementação e na manutenção. A adoção do Modelo de Referência OSI da ISO para definir a arquitetura dos Sistemas Distribuídos introduziu novos conceitos como camadas, protocolos, serviços e interface de comunicação.

O modelo OSI/ISO é estruturado em sete camadas onde cada uma recebe serviços da camada inferior e fornece serviços a camada imediatamente superior. Estes serviços, juntamente com os protocolos que descrevem as regras de comunicação são dois aspectos fundamentais da comunicação em sistemas distribuídos. As especificações de serviços e protocolos podem ser descritas a partir de Técnicas de Descrição Formal (FDT), e permitirão verificar que o protocolo (N) juntamente com os serviços (N-1) fornecem bem o serviço (N) desejado. Tal descrição poderá ser igualmente utilizada em todas as fases posteriores, como implementação e testes.

A interconexão dos diversos equipamentos inteligentes existentes na indústria de manufatura, tornou necessária a utilização de alguns protocolos padronizados das camadas do Modelo de Referência para formar a proposta de arquitetura MAP, que se constitui num padrão internacional para comunicação em ambientes industriais.

Esta solução contribui para a interoperabilidade de equipamentos heterogêneos, possibilitando a integração das mais diversas atividades componentes do ciclo industrial num processo de Manufatura Integrada por Computador (CIM); desta forma são proporcionados o aumento da produtividade e da qualidade dos produtos, requisitos indispensáveis para a competição mundial.

Neste trabalho, atenção especial foi dada ao estudo dos serviços e protocolos MMS, adotados na camada de Aplicação da arquitetura MAP; estes serviços disciplinam a operação cooperativa entre diferentes equipamentos programáveis reduzindo, desta forma, o custo de interoperabilidade dos mesmos.

Para facilitar o entendimento destes serviços e salientar todos os conceitos envolvidos, os mesmos foram posicionados no exemplo de um sistema de produção real, destacando a descrição através de objetos abstratos e do Dispositivo de Manufatura Virtual (VMD), que será, certamente, um modelo geral para os equipamentos de manufatura.

Modelou-se também alguns aspectos de funcionamento do protocolo MMS, de forma que o comportamento do mesmo pudesse ser bem compreendido, permitindo decompor o modelo total do protocolo em níveis de abstração suficientes para um estudo sistemático do mesmo. Entretanto, muitos outros aspectos podem ser ainda modelados para uma futura validação, como por exemplo a inclusão de múltiplas instâncias de pedidos de serviços pendentes, a ocorrência de erros na comunicação, a entrada no ambiente MMS com os parâmetros negociados, a correspondente saída deste ambiente de forma negociada e/ou abrupta em qualquer estado possível, além da determinação de erros de protocolos. As restrições apresentadas aqui são devidas principalmente as ferramentas disponíveis para a análise, porém outras ferramentas com maior potencial de representação (redes de Petri de Alto Nível ou Linguagens de Especificação Formal) deverão ser usadas para permitir a validação em vista de uma implementação correta.

O estudo realizado permitiu então o entendimento dos serviços e do protocolo MMS, como uma primeira etapa necessária para uma posterior utilização. É ainda necessário realizar alguns trabalhos que utilizem estes serviços, além da implementação dos protocolos e serviços fornecidos, para permitir um maior conhecimento do mesmo, que é fundamental para a automação industrial.

BIBLIOGRAFIA

(Ayache 85) AYACHE, J.M. & COURTIAT, J.P. & DIAZ, M. & JUANOLE, G. "Utilisation des réseaux de Petri pour la modélisation et la validation des protocoles". TSI - Techinique et Science Informatigues. vol 4, n.1, 1985.

(Bartoli 83) BARTOLI, P.D. "The application layer of the reference model of open systems interconection". vol.71, n.12, dec 1983.

(Billington 87) BILLINGTON, J. "Protocol Engineering and Nets". may 1987.

(Bochmann 80) BOCHMANN, G.V. & SUNSHINE, C.A. "Formal Methods in Communication Protocol Design". IEEE Trans Com-28, No. 04. Abr. 1980.

(Bochmann 83) BOCHMANN, G.V. "Concepts for Distributed Systems Design". Springer-Verlag 1983

(Bollognesi 87) BOLLOGNESI, T.& BRINKSMA, Ed "Introduction to the ISO Specification Langage LOTOS". Elsevier Science Publishers B.V 1987.

(Bourguet 87) BOURGUET, A. "A Petri Net Tool for Service Validation in Protocol". Protocol Specification Testing and Verification. IFIP,1987.

(Budkoviski 87) BUDKOVISKI, S. & DEMBINSKI, P. "An Introduction to Estelle: A Specification Langage for Distributed Systems". Elsevier Science Publishers B.V 1987.

(Chapin 83) CHAPIN, A.L. "Conections and Conectionless Data Transmission". Proc. of the IEEE. vol.71, no.12. dec 1983.

(Chochon 86) CHOCHON,H. "Programmation et conduite de tâches d´assemblage robotisées: modelisation et processus decisionnels". These de Docteur Ingenieur de l´Université Paul-Sabatier, LAAS - Toulouse 1986.

(Clark 78) CLARK, D.D. & POGRAN, K.T. & REED, D.P. "An introduction to local area networks. Proc of the IEEE. vol.66, no.15. nov 1978.

(Courtiat 87) COURTIAT, J.P. "Contribution a la Description Formelle de Protocoles". These de Docteor d´Etat de L´Université Paull-Sabatier, LAAS - Toulouse. Dec 1987.

(Courtiat 84) COURTIAT, J.P. & AYACHE, J.M. & ALGAIRES,B. "Petri Nets Are Good for Protocols". Computer Communication Review, 14, nc.2. juin 1984.

(Courtiat 86) COURTIAT, J.P. & DEMBINSKI, P. & GROZ, R. &JARD, C. "Estelle: Un Langage ISO Pour les Algcrithmes Distribués et les Protocoles". INRIA. Rapport de Recherche no.595, dec 1986.

(Day 83) DAY, J.D. & ZIMMERMANN, H. "The OSI reference model". Proc. of the IEEE. vol.71, no.12, dec 1983.

(Desclaux 85) DESCLAUX, C. "Melicor: meta-langage instanciable pour da commande repertier". 18 dec 1985.

(Diaz 87) DIAZ, M. "Applying Petri net Based Models in The Design of Systems". LAAS du CNRS. may 1987.

(Diaz 82) DIAZ, M. "Modeling and Analysis of Comunication and Cooperation Protocols Using Petri net Based Models". Proc of the IFIP WG 66.1. 2o. international Workshop on Protocol Specification, Testing, and Verification. May 1982.

(Dickson 83) DICKSON, G.P. & CHAZAL, P.E. "Status of CCITT Description Techiniques and Application to Protocol Specification". Proc of the IEEE. vol.71, no.12, dec 1983.

(Enslow 81) ENSLOW Jr, P.H. "Distributed and Descentralized Control in Fully Distributed Processing Systems".1981.

(Farowich 86) FAROWICH, S.A. "Comunicating in the techinical office". IEEE Espectrum. apr 1986.

(Genrich 79) GENRICH, H.J. & LAUTEMBACH, K. "The Analysis of Distributed Systems by Means of Predicate/Transition Nets". Notes in Computer Sciences, vol. 70, 1979.

(Gomide 86) GOMIDE,F.A. & NETO, M.L.A. "Introdução a Automação Industrial Informatizada". I EBAI 1986.

(Hagar 86) HAGAR Jr, M.L. "The RS-511 manufacturing messaging service nears reality for layer seven". Control Engineering. oct. 1986.

(Holliday 87) HOLLIDAY, M.A. & VERNON, M.K. "A generalized timed petri net model for performance analysis". IEEE trans on software. Eng. vol.SE-13, no.12, dec 1987

(IEC 86) IEC. "Field bus standard for use in industrial control systems". Functional requeriments. Draft for national comment. jul 1986

(Jayasumana 85) JAYASUMANA, A.P. "An Overview of MAP Lower Layer Protocols". IECON. 1985.

(Jensen 81) JENSEN, K. "Coloured Petri Nets and the Invariant Method". Theor. Comp. Science. 1981.

(Kaminski 86) KAMINSKI Jr, M.A. "Protocols for comunicating in the factory". IEEE Spectrum. apr 1986.

(Keller 76) KELLER, R.M. "Formal Verification of Parallel Programs". Com ACM, vol 19, No. 07. Jul 1976.

(Lages 86) LAGES, N.A.C. & NOGUEIRA, J.M.S. Introdução aos Sistemas Distribuídos. Campinas: Editora da UNICAMP/Editora Papirus, 1986.

(Leão 87) LEÃO, J.L.S. & PEDROZA, A.C. "Linguagens para especificação de sistemas distribuídos em automação industrial". Seminário de Automação Industrial. Fpolis. jul 1987.

(Leite 86) LEITE, J.R.E. & FERREIRA, L.R. & MENDES, M.J. "A interconexão de sistemas e automação industrial". Máquinas e metais. - out 1986.

(Leite ...) LEITE, J.R.E. & MENDES, M.J. & MAGALHÃES, M.F. "Protocolos de aplicação em redes locais de computadores na Automação industrial". (Protocolos MAP/TOP). 5o. Simpósio Brasileiro de Redes de Computadores.

(LeLann 82) LE LANN, G. "Le probleme des Systémes temps reel dans le cadre d´architectures distribuées". Convention Informatique septembre 1982, vol.A, 1982.

(LeLann 83a) LE LANN, G. "Trends in Industrial Local Area Networks". Production Management, EIASM, jan 1983.

(LeLann 83b) LE LANN, G. "On Real-Time Distributed Computing". Proc. IFIP Congress 83, Paris. Sept. 1983.

(Linington 83) LININGTON, P.F. "Fundamentals of the Layer Service Definitions and Protocol Specifications". Proceedings of the IEEE. vol.71, no.12, dec 1983.

(Linn ...) LINN Jr, R.J. "The Features and Facilities of Estelle". NBS.

(MAP 87) MAP. "Manufacturing automation protocol" V3.0 Implementation release. apr 1987.

(Mazzola 88) MAZZOLA, V.B. "Estude et Specification do Systeme de Pilotage d´Une CelluleFlexible D´assemblage" DEA Automatique, Informatique industrielle et Traitement do Signal. LAAS/CNRS juin 1988.

(McClelland 83) McCLELLAND, F.M. "services and protocols of the physical layer". Proceedings of the IEEE. vol.71, no.12, dec 1983.

(Mendes 87) MENDES, M.J. & MAGALHÃES, M. "Redes Industriais e o Projeto de Padronização MAP/TOP". SBA Controle e Automação. Vol. 2, No. 1. 1987

(Mendes 86a) MENDES, M.J. "Redes locais de comunicação em ambiente industrial". Máquinas e metais. ago 1986.

(Mendes 86b) MENDES, M.J. & FERREIRA, R.L. & EMILIANO, J.R.L. "Interconexão de sistemas computacionais abertos em automação industrial". SBA: controle e automação. vol 1, No.1. Out. 1986.

(Merlin 85) MERLIN, P.M. "Specification and Validation of Protocols". IEEE trans on comunications. vol com-27, no.11, nov 1979.

(Meyer 85) MEYER, S.P. & MacLeod, T.M. & RODD, M.G. & BLOCH,G. "Real-Time Distributed Computer Control in Flexible Manufacturing Systems". IFAC Distributed Computer Control Systems.1985.

(Pelton 86) PELTON, G.A. "Application Architecture for Communications on the Manufacturing Floor". IECON 1986.

(Prince 81) PRINCE, S.M. & SLOMAN, M. "Comunication Requiriments of a Distributed Computer Comtrol Systems" IEE Proc. vol.128, Pt.E, no.1. jan 1981.

(Rhein 86) RHEIN, V. Von. "Manufacturing Message Specification".1986.

(Saltzer 81) SALTZER, J.H. & CLARK, D.D. "why a ring " IEEE. 1981.

(Schutz 86) SCHUTZ, H.A. "The Role of MAP in Factory Integration". IECON. 1986

(Shapiro 87) SHAPIRO, S.F. "Industrial Networks Werestle With Standards". Computer design. aug 1987

(Souza 87) SOUZA, W.L. "Utilização dos conceitos de módulo, porta e canal em Especificações formais de serviços, Protocolos e Interfaces de comunicação". 3o. Simpósio Brasileiro de Redes de Computadores. 1987.

(Tarouco 86) TAROUCO, L.M.R. Rede de Computadores Locais e de Longa Distância. São Paulo: McGraw-hill, 1986.

(Tomesse 86) TOMESSE, J.P. & DELCUVELLERIE, J.V. "Functional and operative architectures in flexible manufacturing systems" IEEE. 1986.

(TOP 87) TOP. "Technical office protocols". V3.0 Implementation release. apr 1987.

(Tschammer 85) TSCHAMMER, V. & WEWER, W. "Local Area Networks in Real-Time Aplications: Performance Aspects". Copyright IFAC Distributed Computer Control Systems.1985.

(Vissers 86) VISSERS, C.A. & LOGRIPPO, L. "The Importance of the Service Concept in the Design of Data Comunications Protocols". Protocol Specification Testing and Verification. IFIP, 1986.

(Vissers 83) VISSERS, C.A. & TENNEY,R.L. & BOCHMANN, G.V. "Formal Description Techiniques". Proc of the IEEE. vol.71, no.12, dec 1983.

(Williams 84) WILLIAMS, T.J. "The Development of Reliability in Industrial Control Systems". IEEE Micro. Dec. 1984.

(Wood 87) WOOD, G.G. "Surrey of lan´s and standards". Computer Standards & interfaces, 6, 1987.

(Zimmermann 80) ZIMMERMANN, H. "OSI Reference Model- The ISO Model of Architecture for Open Systems Interconnection". IEEE Trans Com. 28, No. 04. Abr. 1980

ANEXO

Descrição dos Serviços MMS

Serviços de Gerenciamento de Contexto

Esta classe de serviços é constituída dos seguintes serviços:

- **"Initiate"**: Este serviço torna possível a um usuário MMS iniciar um diálogo com outro usuário MMS, estabelecendo os recursos necessários para manter a comunicação no contexto MMS, através da negociação dos serviços que serão suportados dentro deste contexto. A execução, com sucesso, do serviço, "Initiate" é indispensável para que qualquer outro serviço possa ser atendido entre um par de usuários MMS. Os argumentos deste serviço incluem o tamanho máximo da mensagem que o sistema deve suportar, o número máximo de pedidos de serviços confirmados pendentes, o nível de aninhamento da estrutura de dados, assim como o CBB horizontal e vertical;

- **"Conclude"**: Este serviço permite a um usuário MMS encerrar a comunicação com outro usuário MMS de forma negociada. É utilizado quando o usuário MMS não tem mais pedidos de serviços a emitir, pois concluiu todos os pedidos de serviços planejados;

- **"Abort"**: Este serviço é utilizado por um usuário MMS para abandonar o contexto MMS imediatamente, de forma abrupta e sem negociação. O provedor de serviço pode também emitir o serviço "Abort", neste caso os dois usuários são comunicados, quando possível, através de uma primitiva de Indicação deste serviço;

- **"Cancel"**: Este serviço permite a um usuário MMS cancelar pedidos de serviços pendentes, isto é, pedidos de serviços que foram emitidos mas que não foram completados através da recepção de uma primitiva de resposta para este serviço. É somente possível para serviços confirmados;

- **"Reject"**: É um serviço inicializado pelo provedor do serviço para notificar a ocorrência de erros de protocolos a um usuário MMS.

Serviços de Suporte do VMD

- **"Status"**: Permite ao usuário MMS determinar as condições gerais da entidade respondedora, através da obtenção do estado do VMD;

- **"UnsolicitedStatus"**: Este serviço, que é do tipo não confirmado, é usado pelo usuário MMS para relatar espontaneamente o seu estado;

- **"GetNameList"**: Permite ao usuário MMS obter a lista (ou parte dela) dos nomes dos objetos definidos no VMD;

- **"Identify"**: Este serviço é usado pelo usuário MMS para obter informações de identificação do usuário MMS respondedor;

- **"Rename"**: Este serviço permite a um usuário MMS modificar o nome de um objeto definido no VMD.

Serviços de Gerenciamento de Domínio

Os domínios são manipulados pelos usuários clientes MMS através de um conjunto de serviços definido no servidor MMS e que realizam operações tais como: a transferência de domínio do cliente ao servidor ("Download"), a obtenção de domínio do servidor pelo cliente ("Upload"), carga pelo servidor de um domínio a partir de um arquivo, o salvamento pelo servidor de um domínio num arquivo, a destruição pelo servidor de um domínio, a determinação dos atributos do domínio. A descrição mais detalhada destes serviços está em anexo.

Os Domínios são manipulados pelo cliente MMS, através dos seguintes serviços definidos no servidor MMS e que operam sobre este objeto:

- **"InitiateDownLoadSequence"**: Permite ao usuário cliente começar o processo de transferência de uma imagem de programa executável para um servidor MMS ("DownLoad");

- **"DownLoadSegment"**: Permite ao usuário servidor obter elementos da imagem carga "Download";

- **"TerminateDownLoadSequence"**: Permite ao usuário MMS terminar o serviço "Download";

- **"InitiateUpLoadSequence"**: Permite a um cliente MMS começar o processo de transferência de uma imagem de programa executável do servidor MMS ("UpLoading")";

- **"UpLoadSegment"**: Permite a um cliente MMS obter um segmento do "UpLoad" do servidor MMS;

- **"TerminateUpLoadSequence"**: Permite ao cliente MMS terminar o processo UpLoad;

- **"RequestDomainDownLoad"**: Permite a um servidor MMS pedir um servidor de arquivo subordinado, para começar uma função "Download";

- **"RequestDomainUpLoad"**: Permite a um servidor MMS pedir um servidor de arquivo subordinado, para começar uma função "UpLoad";

- **"LoadDomainContent"**: Permite a um cliente MMS pedir que o servidor MMS tome a ação de carregar um domínio. Esta carga pode originar do próprio arquivo do servidor, ou pode levar o servidor a pedir serviços de um servidor de arquivo subordinado;

- **"StoreDomainContent"**: Permite a um cliente MMS pedir que o servidorMMS tome a ação de transferir um conteúdo de Domínio a um arquivo (realizar um "UpLoad");

- **"DeleteDomain"**: Permite a um cliente MMS pedir que o servidor MMS delete o domínio especificado e faça os seus recursos disponíveis;

- **"GetDomainAtribute"**: Permite a um cliente MMS pedir que o servidor MMS providencie uma lista dos atributos do domínio especificado;

- **"ObtainFile"**: Permite a um cliente MMS pedir que o servidor MMS tome as ações apropriadas para adquirir um arquivo nomeado para seu arquivo local. Este arquivo nomeado é para ser adquirido ou do cliente MMS ou de um servidor de arquivo subordinado;

Serviços de Gerenciamento de Invocação de Programa

O curso de invocações de programa são direcionados pelo cliente MMS através dos seguintes serviços:

- **"CreateProgramInvocation"**: Este serviço é usado pelo cliente para criar um novo objeto Invocação de Programa em VMD;

- **"DeleteProgramInvocation"**: Este serviço é utilizado pelo cliente para "deletar" um objeto Invocação de Programa em VMD;

- **"Start"**: Este serviço é utilizado pelo cliente para levar uma Invocação de Programa previamente definida no estado "RUNNING";

- **"Stop"**: Este serviço é utilizado pelo cliente para passar uma Invocação de Programa do estado "RUNNING" ao estado "STOPPED";

- **"Resume"**: Este serviço é utilizado pelo cliente para passar uma Invocação de Programa que está no estado "STOPPED" para o estado "RUNNING";

- **"Reset"**: Este serviço é utilizado pelo cliente para passar uma Invocação de Programa de estado "STOPPED" para o estado "IDLE";

- **"Kill"**: Este serviço é utilizado pelo cliente para terminar uma Invocação de Programa, colocando-o no estado "UNRUNNABLE";

- **"GetProgramInvocationAttribute"**: Este serviço é utilizado pelo cliente para determinar os atributos de uma Imvocação de Programa, seu estado, sua lista de domínios dependentes, e seu tempo de vida.

Serviços de Acesso Variável

Serviços usados pelo MMS cliente:

- **"Read"**: Para ler o conteúdo de uma ou mais variáveis MMS;

- **"Write"**: Para atualizar o conteúdo de uma ou mais variáveis MMS;

- **"InformationReport"**: Pedido pelo usuário cliente ou usuário servidor (VMD) para informar outro usuário MMS (servidor ou cliente) o valor de uma ou mais variáveis especificadas, como lido pelo usuário MMS requisitor;

- **"GetVariableAcessAtributes"**: Retorna os atributos de um objeto de acesso à variáveis do VMD;

- **"DefineNamedVariable"**: Criam um objeto variável nomeado no VMD;

- **"DefineScatteredAcess"**: Cria um objeto de acesso "Scattered", cujos componentes são objetos variáveis nomeadas, não-nomeadas, ou mesmo acesso "scattered";

- **"GetscatteredAcessAtributes"**: Retornar os atributos de Objeto de Acesso "scattered";

- **"DeleteVariableAcess"**: Par delir um ou mais objetos de acesso a variável nomeada ou "Scattered";

- **"DefineNamedVariableList"**: Cria um objeto Lista de Variável nomeada;

- **"GetNamedVariableListAttributes"**: Para pedir os atributos de um objeto lista de variável nomeada definido no VMD;

- "**DefineNamedType**: Para armazenar uma especificação do tipo nomeado para uso em definições posteriores de variáveis ou tipos;

- "**GetNamedTypeAttributes**": Para pedir os tributos de um objeto tipo nomeado;

- "**DeleteNamedType**": Para delir um ou mais objetos tipo nomeado.

Serviços de Gerenciamento de Semáforo

Estes serviços permitem a definição e o controle de semáforos, no MMS há duas classes genéricas de semáforos:

- Semáforo "ficha" ("token"): Permitindo simples ou múltiplos proprietários;

- Semáforo "Pool": Permitindo uma alocação dinâmica ou explícita de fichas nomeadas;

Um Semáforo é modelado como um servidor, uma lista de proprietários e uma fila de requisitores. A diferença entre os dois tipos de Semáforos é a identidade das fichas, que existe no semáforo "pool" através de um nome individualizado para cada ficha.

Os seguintes serviços são usados pelos usuários MMS para operarem sobre esses objetos:

- "**TakeControl**": Permite obter o controle de um semáforo;

- **"RelinquishControl"**: Libera o controle de um semáforo obtido;

- **"DefineSemaphore"**: Cria um semáforo "Token" no usuário MMS respondedor;

- **"DeleteSemaphore"**: Permite delir um semáforo;

- **"ReportSemaphoreStatus"**: Permite obter o estado sumarizado de um semáforo;

- **"ReportPoolSemaphoreStatus"**: Permite obter o nome e o estado de "Tokens"nomeados controlado pelo semáforo "Pool";

- **"ReportSemaphoreEntryStatus"**: Permite obter o estado detalhado da lista de proprietários e requisitores relacionados a um semáforo.

Serviços de Comunicação com o Operador

São serviços que permitem a comunicação com o operador da estação através de saída e entrada de dados. Para isso é utilizado o objeto Estação Operador que descreve como trabalham os seguintes serviços:

- **"Input"**: Permite a um usuário MMS obter dados da estação operador, com opção de ecoá-los no terminal de saída;

- **"Output"**: Permite a um usuário MMS escrever uma mensagem para a estação operador.

Serviços de Gerenciamento de Eventos

- **"DefineEventCondition"**: Para criar um objeto condição de eventos;

- **"DeleteEventCondition"**: Para delir um ou mais objetos de condição de Eventos definidos no MMS;

- **"GetEventConditionAttributes"**: Permite obter os atributos descritivos de uma condição de evento;

- **"ReportEventConditionStatus"**: Para obter o estado de uma condição de evento;

- **"AlterEventConditionMonitoring"**: Para alterar qualquer combinação dos atributos **de** uma condição de evento monitorada, como "Enable", "Priority" e "Alarm Sunmary Reports";

- **"TriggerEvent"**: Para disparar um evento associado com uma condição de evento;

- **"DefineEventAction"**: Para criar um objeto ação de evento num VMD;

- **"DeleteEventAction"**: Para delir um ou mais objetos Ação de Evento definidos em um VMD;

- **"GetEventActionAttributes"**: Para obter os atributos de uma ação de eventos definidos no VMD;

- **"ReportEventActionStatus"**: Para obter o número de registros de eventos que só especificam uma ação de evento definido no VMD;

- **"DefineEventEnrollment"**: Para que o VMD adicione o cliente requisitor ou um cliente subordinado, na lista de usuários para os quais os pedidos de serviços "EventNotification" resultante de uma condição de evento especifico estão para ser enviados;

- **"DeleteEventEnrollment"**: Para delir um ou mais Registros de Eventos;

- **"GetEventEnrollmentAttributes"**: Para uma lista de notificação de registros de eventos que satisfazem um conjunto especificado decritérios;

- **"ReportEventEnrollmentStatus"**: Para obter o estado de uma simples notificação de registro de evento;

- **"AlterEventEnrollment"**: Para recolocar os atributos "EventConditionTransitions" e/ou "AlarmAcnowLedEventRule" de um objeto Registro de Evento referenciando na condição de evento monitorada;

- **"EventNotification"**: É um serviço não confirmado utilizado para notificar a um cliente registrado da ocorrência de uma transição de estado associado com a condição de evento;

- **"AcknowledgeEventNotification"**: Para notificar um VMD que o usuário reconheceu uma notificação de evento recebido do VMD;

- **"GetAlarmSumary"**: Para obter informações do VMD sobre o estado corrente de condições de eventos com o atributo "AlarmSummaryRequest" verdadeiro;

- **"GetAlarmEnrollmentSumary"**: Para obter um resumo do VMD sobre o estado corrente de um registro de evento tendo uma "AlarmAcknowLedgmentRule" diferente de "None".

Serviços de Gerenciamento de Ocorrências (Jornal)

Serviços usados pelo usuário MMS cliente:

- **"ReadJournal"**: Para recuperar as informações de uma ocorrência especificada;

- **"WriteJournal"**: Para atualizar as informações no registro de ocorrências;

- **"InitializeJournal"**: Para iniciar todo ou parte de um registro de ocorrências ao estado de vazio;

- **"ReportJournalStatus"**: Para determinar o número de entradas em um registro de ocorrências;

Serviços de Gerenciamento de Arquivos

Estes serviços permitem a manipulação de pequenos arquivos seqüenciais em aplicações industriais, transferindo de/para VMD, de forma mais simples que o protocolo FTAM da ISO.

- **"FileOpen"**: Permite abrir um arquivo para ser lido;
- **"FileRead"**: Para transferir parte ou todo conteúdo de um arquivo aberto do servidor para o cliente MMS;
- **"FileClose"**: Permite liberar os recursos associados com a transferencia do arquivo;
- **"FileRename"**: Permite trocar o nome de um arquivo virtual no servidor MMS;
- **"FileDelete"**: Permite delir um arquivo do servidor MMS;
- **"FileDirectory"**: Permite obter os nomes e os atributos de um arquivo ou grupo de arquivos no servidor MMS.